UNIVERSITY OF CALIFORNIA
SANTA CRUZ

CURATING SIMULATED STORYWORLDS

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTATIONAL MEDIA

by

James Ryan

December 2018

The Dissertation of James Ryan
is approved:

Michael Mateas, Chair

Noah Wardrip-Fruin

Ian Horswill

Jonathan Lessard

Lori Kletzer
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Curating Simulated Storyworlds

by

James Ryan

There is a peculiar method in the area of procedural narrative called *emergent narrative*: instead of automatically inventing stories or deploying authored narrative content, a system simulates a storyworld out of which narrative may emerge from the happenstance of character activity in that world. It is the approach taken by some of the most successful works in the history of computational media (*The Sims*, *Dwarf Fortress*), but curiously also some of its most famous failures (Sheldon Klein's automatic novel writer, *Tale-Spin*). How has this been the case? To understand the successes, we might ask this essential question: what is the pleasure of emergent narrative? I contend that the form works more like nonfiction than fiction—emergent stories actually happen—and this produces a peculiar aesthetics that undergirds the appeal of its successful works. What then is the pain of emergent narrative? There is a ubiquitous tendency to misconstrue the raw transpiring of a simulation (or a trace of that unfolding) as being a narrative artifact, but such material will almost always lack story structure.

So, how can the pain of emergent narrative be alleviated while simultaneously maintaining the pleasure? This dissertation introduces a refined approach to the form, called *curationist emergent narrative* (or just *curationism*), that aims to provide an answer to this question. Instead of treating the raw material of simulation as a story, in curationism that material is *curated* to construct an actual narrative artifact that is then mounted in a full-fledged media experience (to enable human encounter with the artifact). This recasts story generation as

an act of recounting, rather than invention. I believe that curationism can also explain how both wild successes and phenomenal failures have entered the oeuvre of emergent narrative: in successful works, humans have taken on the burden of curating an ongoing simulation to construct a storied understanding of what has happened, while in the failures humans have not been willing to do the necessary curation. Without curation, actual stories cannot obtain in emergent narrative.

But what if a storyworld could curate itself? That is, can we build systems that *automatically* recount what has happened in simulated worlds? In the second half of this dissertation, I provide an autoethnography and a collection of case studies that recount my own personal (and collaborative) exploration of automatic curation over the course of the last six years. Here, I report the technical, intellectual, and media-centric contributions made by three simulation engines (*World*, *Talk of the Town*, *Hennepin*) and three second-order media experiences that are respectively driven by those engines (*Diol/Diel/Dial*, *Bad News*, *Sheldon County*). In total, this dissertation provides a loose history of emergent narrative, an apologetics of the form, a polemic against it, a holistic refinement (maintaining the pleasure while killing the pain), and reports on a series of artifacts that represent a gradual instantiation of that refinement. To my knowledge, this is the most extensive treatment of emergent narrative to yet appear.

To Nina, who makes it possible

Acknowledgments

I cannot imagine a more fostering intellectual environment than the Expressive Intelligence Studio at UC Santa Cruz, and this is due in no small part to those whom I have been so fortunate to call my labmates: Matthew Balousek, Bryan Blackford, Sherol Chen, Kate Compton, Melanie Dickinson, Cyril Focht, Jacob Garbe, John Grey, April Grow, Nick Junius, Eric Kaltman, Max Kreminski, Paul Maddaloni, Josh McCoy, Marcelo Viana Neto, Stacey Mason, John Mawhorter, Peter Mawhorter, Josh McCoy, Wes Modes, John Murray, Joe Osborn, Aaron Reed, Ben Samuel, Ben Spalding, Adam Summerville, and Dietrich Squinkifer.

More broadly, I owe so much to the larger ecosystem surrounding computational media and computer science at UC Santa Cruz, and namely to the people who have made it so great: Rob Abbott, Nathan Alice, Morteza Behrooz, Chandranil Chakraborttii, Ryan Compton, Peter Cottrell, Jared Duval, Daniel Fava, Angus Forbes, Sarah Harmon, Chao Hu, Hope Hutman, Arnav Jhala, Katherine Isbister, Dylan Lederle-Ensign, Grace Lin, Daniel Lipovetsky, Emily Lovell, Stephanie Lukin, Isaac Karth, Mitch Mastroni, Jo Mazeika, Eddie Melder, Amita Misra, Afshin Mobramaein, Jim Moore, Johnathan Pagnutti, Elahe Rahimtoroghi, Allie Riggs, Elena Rishes, Joe Rossi, Jen Sawyer, Marcelo Siero, Adam Smith, Dhanya Sridhar, Michael Thomét, Jim Whitehead, and Jimmy Zhou.

It was incredibly rewarding to mentor, and collaborate with, a multitude of undergraduate researchers across a variety of projects during my time at UC Santa Cruz: Chase Armer, Casey Barackman, Tyler Brothers, Janel Catajoy, Aislynn Cetera, Lisa Durand, Andrew Max Fisher, Tim Hong, Yani Mohamad Fauzi, Trevor Holoch, Nick Kontje, Adesh Kumar, Merita Lundstrom, David Nguyen, Jacinda Ni, Jinah Noh, Jared Ono, Taylor Owen-Milne, Silvia Ordonez, Tiffany Phan, Sergiy Ravnyago, Emily Rodriguez, Joyce Scalettar, Ethan Seither, Thomas

Ruiz, Thovatey Tep, Kevin Yeh, Jill Z. Yeung, Reshma Zachariah, and all the unforgettable students in the 2016–17 18X sequence.

To the countless friends and acquaintances I've made in this field: there are too many of you to name here, but know that I have appreciated my time with you. I must identify a few individuals, however—thanks for the friendship and intellectual exchange, Tarn Adams, Mike Cook, Bill Ferguson, Swen Gaudl, Hartmut Koenitz, Ben Kybartas, Vincenzo Lombardo, Mark Nelson, Antonio Pizzo, and Steve Wingate.

I had critical mentors in my academic life before arriving at UC Santa Cruz, without whom I'd be somewhere else: Johan Christopherson, Genevieve Melton-Meaux, and Serguei Pakhomov, you are beacons in the night.

I would like to extend special gratitude to Marilyn Walker, who took a flier on me when I applied to a computer science PhD program no formal background whatsoever in the field. Without you, I would not have made it to Santa Cruz. Additionally, I should specifically identify Reid Swanson, who was like a fourth advisor for me when he worked here as a postdoc.

To the reading committee of this dissertation—Ian Horswill, Jonathan Lessard, Michael Mateas, and Noah Wardrip-Fruin—thank you endlessly for enduring this tome and for providing invaluable feedback and support along the way. Ian, I met you at my first conference in this field, and you've been a trusted mentor, friend, and fellow Minnesotan for my entire PhD tenure. Jonathan, you are so effortlessly keen and affable; to many more conversations! My gratitude extends also to David Elson, who served on the reading committee for my thesis proposal, and whose ideas persist still with regard to this new topic.

To Michael and Noah: wow. Your imprint is on this document and on my life. I absorbed so much in a relatively short while; I had such a great time. I'm proud to have operated under your tutelage and in the intellectual tradition of the Expressive Intelligence Studio, a wonderland you made for us.

To my parents, grandparents, siblings, extended family, and friends: thank you for believing in me and for reminding me of more important things.

Finally, Nina: no words on a page are enough—you are everything.

Chapter 1

Introduction

Dwarf Fortress is beloved for the stories that emerge out of its simulated storyworlds. It has been called the most complex videogame ever [1321, 1281, 219, 306, 244, 1125], and that complexity is always in the service of emergent narrative intrigue. The scholars Stepahie Boluk and Patrick LeMieux put it succinctly: "*Dwarf Fortress* begins and ends with stories" [126, p. 133]. As a fitting testament to its complexity, someone has written an entire book explaining how to play the game: the remarkable *Getting Started with Dwarf Fortress: Learn to Play the Most Complex Video Game Ever Made* [1281]. In its afterward, author Peter Tyson identifies the game's narrative heart:

> The joy of Dwarf Fortress lies in the game's remarkable depth and the stories that unfold from the interactions between its many elements. From world and history generation through to the individual personalities of your dwarves, each component of Dwarf Fortress involves us in amazing stories making each fortress a unique one. By now, I hope you too have a few stories to share [1281, p. 197]

Dwarf Fortress players famously do share their stories [261], on web forums [522], dedicated websites [1287], in magazines [255], and elsewhere. *The Sims* players share stories too [150, 527, 27, 26], as do players of other games [867].

This centrality of narrative in *Dwarf Fortress* has been articulated by cocreator Tarn Adams himself, who has called the system a *story generator*, as the writer Jonah Weiner recounts in a feature on the game that appeared in *The New York Times Magazine*:

> Many video games mimic the look and structure of films: there's a story line, more or less fixed, that progresses only when you complete required tasks. This can make for gripping fun, but also the constrictive sense that you are a mouse in a tricked-out maze, chasing chunks of cheese. Tarn envisions Dwarf Fortress, by contrast, as an open-ended "story generator." [1321, n.p]

So what kind of story generator is this? In the field of artificial intelligence, computational story generation has a history dating to at least 1960 [1039], but many systems in the area do not work like this. More often, they tell stories by modeling a top-down process of plot invention [666, 928, 391, 1163, 816, 1004] or creative writing [1276, 936, 937]. But *Dwarf Fortress* takes a bottom-up approach: a simulated storyworld teems with simulated characters living simulated lives, and when stories are created by the system, it is through the happenstance of this character simulation.[1] This is called *emergent narrative*, which brings me to the first research question that is tackled in this dissertation:

RQ1. What is emergent narrative?

In emergent narrative, stories arise bottom-up through the mechanism of computer simulation. Often this means a simulation that undergirds an interactive experience, but emergent narrative can also obtain in noninteractive contexts,

[1] To be clear, by 'storyworld', I mean a world, real or otherwise, whose events are narratable. This could be a fictional world that is recounted in a work of literature, or a simulated world out of which narrative may emerge. Of course, the latter sense of the term is most pertinent to the concerns of this thesis. If this term feels redundant or unnecessary, given 'world', then hold on for now—later on, I will discuss examples of computer simulations that do not seem capable of producing narrative (more precisely: generating events that may be narrated).

such as with simulationist story generators. In Chapter 2, I handle this research question by providing an extensively deliberated definition of the form, a first published account on the origins of the term 'emergent narrative', and a brief overview of academic work in the area to date.

But why is this even important? Emergent narrative is important because it represents arguably the most successful approach to procedural narrative in the public consciousness, through its utilization in popular videogames such as *Dwarf Fortress* and *The Sims*. This success leads me to my second research question:

RQ2. What is the pleasure of emergent narrative?

Clearly there is a distinct pleasure constituted in works of emergent narrative, but what exactly is it and what are its contributing factors? While researchers have written extensively about the merits of emergent narrative as an approach to interactive storytelling [50, 701, 702, 703, 55, 1233, 617, 1230], I am not aware of any substantial treatise on the distinct experiential pleasures of the form. In Chapter 3, I attempt to unpack the success of emergent narrative by discussing the approach in terms of a series of analogies to other forms, namely nonfiction, which work to yield a distinctive set of aesthetics that I call the *aesthetics of emergent narrative*. I believe that the pleasure of the form is rooted in these analogies and characterized by these aesthetics, and with this chapter I aim to provide the most extensive apologetics for emergent narrative to date.

But while it arguably represents the most successful approach to procedural narrative in the public consciousness, emergent narrative has characterized academic story generation's most famous failures, namely Sheldon Klein's murder-mystery generator [597] and Jim Meehan's *Tale-Spin* [822]. In the decades following those early systems, scholars and practitioners in the area have generally come to view the approach as at best a dead end, namely for the reason that emergent narrative tends to lack story structure. Here, an excerpt from a paper by *Façade* [775] creators Michael Mateas and Andrew Stern expresses this critique:

> But for believable agents used as characters in a story world, strong autonomy becomes problematic. Knowing which action to take at any given time depends not just on the private internal state of the agent plus current world state, but also on the current story state, including the entire past history of interactions building on each other towards some end. The global nature of story state is inconsistent with the notion of an autonomous character that makes decisions based only on private goal and emotion state and local sensing of the environment. [774, p. 223]

Mateas, who is my coadvisor, has proffered a more compact critique of emergent narrative in conversation: *it's just one damn thing after another*. While Ruth Aylett, Sandy Louchart, and collaborators—who carry the term 'emergent narrative' as a banner—have worked to cultivate the approach in the specific domain of interactive storytelling [50, 701, 55], its critics have generally remained skeptical [776, 1006, 330, 1003, 1004, 1372, 1013]. This raises my third research question:

RQ3. What is the pain of emergent narrative?

In Chapter 4, I unpack a series of harsh critiques of emergent narrative—namely ones proffered by Espen Aarseth and Noah Wardrip-Fruin—to identify a series of pitfalls that have worked to hamper the form. These hazards operate at both the level of underlying simulation and the level of surface expression, working to inhibit either the emergence of story structure or its identification by human observers (or both). This is the pain of emergent narrative, and with this chapter I aim to provide the most brutal critique of the form to date.

With this double-edged foundation, we arrive at the fundamental research question that is tackled in this dissertation:

⋆ **RQ4. How can the pain of emergent narrative be alleviated while simultaneously maintaining its pleasure?**

One potential treatment for the pain is to *intervene* in the storyworld to remediate its affairs such that story structure reliably obtains in each simulation instance. Such intervention characterizes technical schemes like *drama management* [758, 1010] and *plot-level narrative planning* [1370, 1004], which employ omniscient agents that dramatically remediate a storyworld according to policies about story structure. But while such remedies may alleviate the pain of emergent narrative, they in turn kill the pleasure: when a simulated storyworld is polluted through the intervention of an external system, events no longer emerge, but instead spawn according to the policies of the modulating system. When this happens, emergent narrative ceases to manifest, and its distinctive pleasures disappear in turn.[2]

This dissertation identifies an alternative mechanism that alleviates the pain of emergent narrative while simultaneously maintaining its pleasure: *curation*. Instead of intervening in a storyworld to *impose* story structure, the world is curated to *identify* stories that authentically emerged. The idea here is that well-crafted

[2]At its heart, this dissertation is an art manifesto that promulgates a fiercely *emergentist* approach to procedural narrative. It is written on behalf of a contingent that appreciates simulated worlds as real places where tiny abstract characters live out tiny abstract lives—this does not necessarily reflect an ontological position, but more essentially an *aesthetic orientation*. As part of a rhetorical strategy for promoting this orientation, in this document I use value-laden terms like 'pure', 'polluted', 'tampered', and more to describe a variety of architectural patterns that are probably equally valid from an intellectual and technical standpoint. As such, my charged language is meant to instantiate this artistic position, rather than to express technical or intellectual appraisal. That is, if I for instance decry an architectural pattern that 'pollutes' a storyworld, I mean to do so purely in terms of the aesthetic orientation for which this dissertation serves as a loose manifesto—even though such a pattern may be incompatible with this aesthetic orientation, as cued by my usage of value-laden terms, it could perfectly valid from any other standpoint. I believe that variation in an underlying architecture yields variation in terms of what is aesthetically possible at the surface level of a work of computational media—this is what Michael Mateas has termed the *interpretive affordances* of an architecture [762]—and in this dissertation I introduce an architecture that was formulated expressly to target a specific set of interpretive affordances, which in turn anchor an intellectual framework that is the primary subject of this work. I will periodically return to these concerns below, but for now you have been warned: this is an emergentist manifesto.

storyworlds do not actually lack narrative intrigue, but rather narratively potent material tends to be embedded in larger accumulations of unremarkable simulated material. This means that emergent narrative may actually have story structure, but the simulated material composing such structure may be embedded in a larger accumulation that itself lacks story structure. Thus, when human players tell stories about what has happened in *Dwarf Fortress* or *The Sims* gameplay, they are in fact carrying out a process of curation—narratively potent material is extracted and then assembled into narrative artifacts with discernible story structure.

What if a system did this automatically? This is the impetus for the subject of this dissertation: *curationist emergent narrative* (or *curationism*, for short). This framework represents a refined approach to emergent narrative that entails the curation of raw simulated material into actual narrative artifacts that are mounted into full-fledged media experiences. This procedure is instantiated in a generalized curationist system architecture, which is itself inspired by Hayden White's model of historiography as a constructive storytelling procedure [1332]. In the architecture, a module called the *chronicler* inscribes the transpiring phenomena of a simulated storyworld in a *chronicle*, which a *story sifter* sifts through to extract material that a *narrativizer* uses to construct a *story*, which is finally mounted into an actual *media experience*, thereby enabling human encounter. This makes story generation a process not of *inventing* stories, but of *recounting* stories about what has actually happened in a world. In Chapter 5, I discuss this framework in terms of its intellectual position, technical approach, generalized architecture, and variants of that architecture.

In the process of outlining the curationist approach in that chapter, I incidentally handle another research question that is implied by the previous ones:

> **RQ5. How has emergent narrative been arguably the most successful approach to procedural narrative and at the same**

time perhaps its least successful approach?

I contend that curationism actually accounts for how certain works of emergent narrative have been tremendously successful, while others have been outright failures. Again, the answer is simply this: curation. Projects like *Dwarf Fortress* and *The Sims* have flourished because human players are willing to curate their simulated storyworlds so that manifest emergent narrative may be encountered (thereby unlocking the aesthetics of the form). Such curation may result in genuine narrative artifacts [1287, 27], which may be mounted in full-fledged media experiences [255, 150], or it may take the form of a kind of unconscious narrativization that drives the basic interpretative procedures that humans employ to make sense of an ongoing simulation (or any experience). I call the latter phenomenon *mental curation*, and in Section 5.3.2 I show that it instantiates a degenerate variant of the curationist architecture.

On the other hand, when human observers are *not* willing to do curation, a work of emergent narrative will fail miserably. For instance, Sheldon Klein's much derided story generator produces monolithic reports on all the various happenings of its underlying simulation, and the result is a jumbled mess that humans have typically been unwilling to curate into actual emergent narrative (thereby precluding the aesthetics of the form). Thus, success in emergent narrative has typically been contingent on whether or not humans are willing (or enabled) to curate a simulated storyworld. In the case of videogames like *Dwarf Fortress* and *The Sims*, humans have largely been willing because, through gameplay, they develop attachments to the storyworld and its characters. This is not typically the case in simulationist story generation, and so we find a generally negative appraisal of that approach. But what if systems automatically curated their own storyworlds?

So far I have outlined the first half of this dissertation, *Part I: Why and*

How, which is about why curationism is a viable framework and how it might be instantiated in actual projects. In total, this half represents what is to my knowledge the most extensive (and most comprehensive) treatise on emergent narrative to date. The second half of this dissertation, *Part II: How and Why*, approaches curationism from the opposite direction—this half is about how I have instantiated the curationist method in my own work and why it appears to have been a good approach. In this way, it wrangles another essential research question:

RQ6. What would a curationist project actually look like?

Given my articulation of the curationist approach, such a project would be sure to incorporate a simulated storyworld, curation, and a media experience. As I explain while outlining variants of the curationist architecture in Section 5.3.2, the media experience and storyworld may be coextensive or decoupled, and curation may be carried out by one or both in either an online or offline manner.

I have spent the last six years developing a series of simulation engines—*World*, *Talk of the Town*, and *Hennepin*—that are intended to drive *second-order media experiences* set in their simulated storyworlds. These engines are each the subject of a dedicated technical chapter in the second half of this dissertation. In tandem with the development of these engines, I have worked independently and in collaboration to develop a series of experiences set in the worlds that are created by those systems. Following each chapter reporting a simulation engine, I provide another, in tandem, that outlines an experience that was driven by the engine. These experiences are: *Diol/Diel/Dial*, a hypertext world encyclopedia that compiles the massive total history of a *World* world; *Bad News*, a collaboration with Ben Samuel and Adam Summerville that takes the form of an installation-based immersive experience (with a live actor) set in a *Talk of the Town* town; and *Sheldon County*, an ongoing project concerning the procedural generation of narrative podcasts each recounting a unique *Hennepin* storyworld. While the simulation chapters report technical contributions, the experience chapters are meant primarily as case studies that concern the instantiation of curationism in the experience at hand (or the fatal lack thereof). In turn, they answer my sixth research question by providing a troubling example of missing curation, a powerful example of human-powered curation, and a first exploration of fully automatic curation. Moreover, through the articulation of a set of actionable strategies that I have developed both in my simulation engines and the media frameworks that curate their generated material, this series of chapters answers the following additional research questions:

> **RQ7. What are some design patterns for crafting simulated storyworlds in ways that facilitate curation?**
>
> **RQ8. What are some design patterns for curating simulated storyworlds?**

I should include a brief note on the research methods employed in this dissertation. As for Part I, my method is scholarly argument in the humanist style, which means I have succeeded to the degree that you are convinced. Like I have mentioned, technical contributions are reported in Part II, but I do not personally see much value in them. As others have articulated before me, computational techniques that are intended to support media experiences cannot be truly appraised except through actual implemented experiences of the kind that they are meant to support [762, 1200, 505, 610, 1316]. From this orientation, technical contributions in computational media are meaningless, and in fact not contributions at all, when they are divorced from the media contexts for which they are intended. In this dissertation, the real appraisals of my simulation engines appear in the case studies of the media experiences that they enable, since it is only in those contexts that the technical approaches taken by the engines can be evaluated. As such, my primary evaluation scheme in Part II is the *case* [103], and in fact this answers the call of Ian Horswill, who is on the reading committee for this dissertation:

> computer science is traditionally, and understandably, resistant to work on cases. One cannot generally get a Ph.D. for writing a program; one gets a Ph.D. for developing a general theory that will help others write some class of programs. But I see no alternative but to relax this constraint for AI work that has specifically aesthetic goals. We have to be open to the case because the case is all we have. [505, p. 84][3]

As I will make clear throughout the rest of this document, the curationist framework of Part I actually coevolved along with the simulation and media practice that I recount in Part II. In this way, and in a certain sense, Part I tells a story about Part II, which itself tells a story about Part I. As I was preparing to write this dissertation, Jonathan Lessard, another member of my reading committee, encouraged me to outline the evolution of my practice using the method of *autoethnography* [310]. I took that to heart, and as a result each chapter in Part II begins with a personal recollection of the intellectual and technical contexts in which the subject of that chapter was developed. As such, this dissertation wrangles a final research question, and it is perhaps the most difficult one:

RQ9. How did I get here?[4]

[3]This quotation originates in a paper titled “Science Considered Harmful”, to which R. Michael Young has recently written a rebuttal titled “Science Considered Helpful” [1371]. The dialogue affords an illuminating look into the nascent field of computational media, as it attempts to define itself.

[4]While I include this somewhat lightheartedly, I do think it is a serious question, especially in the context of a PhD thesis. As a digger in the historical record of computational media, I have read through many dissertations that were published decades ago. When I do so, I am generally more interested in the intellectual contexts in which the student carried out the subject project than in the technical and scientific details that are reported in the document. As such, acknowledgments sections are frequently more potent (from a contemporary intellectual standpoint) than dissertation bodies, because in the latter there is a tendency to eschew any recounting that may be perceived as personal. When a scholarly work lacks contextual informa-

This is the first ever dissertation in computational media. The Computational Media PhD program at UC Santa Cruz started up last year [1197] as an offering of the Department of Computational Media, which was itself established in 2014 [1196, 609]. Prior to that, a group of leading scholars in the field convened on UC Santa Cruz in the summer of 2012 under the moniker of the *Media Systems* project.[5] In "Envisioning the Future of Computational Media: The Final Report of the Media Systems Project", my coadvisors Noah Wardrip-Fruin and Michael Mateas articulate how work in this emerging area weaves technical, creative, interpretative, and collaborative threads to produce four kinds of outcomes:

> *Artifacts*: the outcome of making novel computationally-driven media.
>
> *Capabilities*: the outcome of developing computational, representational, and design approaches that enable new forms of media.
>
> *Insights*: the outcome of studying the technical, historical, and cultural creation and function of computationally-driven forms of media, both old and new.
>
> *Educated Practitioners*: the outcome of interdisciplinary education and training in computational media. [1316, p. 21].

In turn, the requirements for this nascent PhD program stipulate that a dissertation in this area should utilize these various modes of work in turn (or in tandem) to produce outcomes of the sorts just listed:

> The Ph.D. dissertation is a significant contribution to the CM field. It may emphasize technical, media-creation, or interpretive aspects, but must have a significant secondary contribution from at least one of the other areas—and may also emphasize areas equally.

This dissertation is the result of technical, creative, interpretative, and collaborative work that has resulted in a series of artifacts, capabilities, and insights, as

tion concerning its own development, it will be at best incomplete and at worst intellectually dishonest, and so in this thesis I seek to provide such information freely and abundantly. While all dissertations are personal narratives, this one is more personal than most.

[5] A project website is hosted here: `https://mediasystems.soe.ucsc.edu`.

well as one educated practitioner. To be more specific, in this document I provide the following contributions:

- *A definition of emergent narrative.* This definition is arrived at through an extensive deliberative process, and it is meant to join three distinct traditions that I view as instantiating the same essential form: the simulationist school of noninteractive story generation, as exemplified by systems like *Saga II*, Sheldon Klein's murder-mystery generator, and *Tale-Spin*; videogames driven by emergent character simulation, such as *Dwarf Fortress* and *The Sims*; and the bottom-up approach to academic interactive storytelling that is exemplified by the 'emergent narrative' research programs of Ruth Aylett, Sandy Louchart, Mariët Theune, Ivo Swartjes, and their respective collaborators. This contribution appears in Chapter 2, though my inclusive treatment of both interactive and noninteractive works is reinforced throughout this document.

- *An apologetics for emergent narrative.* While others have argued extensively about the merits of interactive emergent narrative, in this document I work to unpack the essence of character simulation itself, which results in the identification of analogies to nonfiction, lived experience, worldbuilding, and art brut. Here, in particular, I spend considerable effort arguing how emergent narrative actually works more like nonfiction than fiction, which I do by reviewing an extensive array of definitions of nonfiction that I then apply to a work of curationist emergent narrative—Tim Dinee's *Oilfurnace* [255]—to show that the definitions are satisfied. Finally, I identify a set of aesthetics that are yielded by these analogies, which I call the *aesthetics of emergent narrative.* This contribution appears in Chapter 3.

- *A polemic against emergent narrative.* In addition to providing what is perhaps the most extensive defense of emergent narrative, I also contribute what may be the harshest attack on the form to date. In this document, I unpack and extend the criticisms of emergent narrative proffered by various writers including Espen Aarseth and Noah Wardrip-Fruin to identify a series of pitfalls at the levels of both simulation and curation. As to the former, I primarily articulate how the crafting of character simulations often fails to target emergent structures that have the discernible features that humans associate with narrative. In a section on curation issues, I unpack Aarseth's denouncement of a troubling (and ubiquitous) conflation of simulation and narration—the raw transpiring of simulation (or a trace of that unfolding) is called a story—to assert that actual emergent narrative can only manifest through a procedure of curation. This contribution appears in Chapter 4.

- *A refined approach to emergent narrative.* This comes in the form of my framework for curationist emergent narrative, which is the subject of this dissertation. In producing this contribution, I proceed from the material contained in both my apologetics and my polemic for the form. Here, my challenge to myself was to devise a refined approach to emergent narrative that takes its harshest appraisals as an intellectual starting point. This is the primary contribution of this dissertation, but Chapter 5 is specifically devoted to the curationist framework.

- *A set of artifacts, in the form of three simulation engines and three media experiences.* These are the completed simulation engines *World* and *Talk of the Town*, the completed media experiences *Diol/Diel/Dial* and *Bad News*, and the in-process engine *Hennepin* and in-process experience *Sheldon County.* These systems and experiences are each the subject of dedicated chapters that appear throughout Part II of this dissertation.

- *A set of case studies in curationist emergent narrative.* For each of the three media experiences that I report, I provide a case study that contemplates the experience in terms of the curationist framework that is the subject of this dissertation. In total, these case studies provide an actionable account both of how curationism may be carried out and why that is worth doing. As to the former, I identify a series of design patterns at the levels of both simulation and curation.

- *An autoethnography of my simulation and media practice.* Across the openings of the chapters in Part II, I tell the story of how my simulation and media practice evolved over time. Incidentally, these case studies also explain how the curationist framework at the heart of this dissertation was developed, since all of this coevolved in the Petri dish of my life. This is a contribution because it elaborates on the core contributions of this thesis by explaining why I made the decisions that I made, why the decisions were good ones or bad ones, and critically how I came to appraise them later on.

- *A history of emergent narrative.* For the last year, I have been carrying out a massive project involving the excavation of materials pertaining to the forgotten history of computational media.[6] While this dissertation is not an historical study, this has been my mindset recently, and so inevitably I provide historical context along the long path of this document. This includes a tracing of the origins of the term 'emergent narrative', which I do not believe anyone has provided before; a first substantial account of the earliest known work of emergent narrative, *Saga II* (1960); an excavation of Sheldon Klein's murder-mystery generator, working in the style of Noah

[6]My first outputs from this project are a conference paper on a forgotten 1960s story generator [1039] and a workshop paper that provides an extensive account of a 1578 work of interactive branching narrative [1041].

Wardrip-Fruin's study of *Tale-Spin* to show how the system has for decades been misunderstood; and numerous details here and there on the history of computational narrative and computational media, more broadly.

With this thesis being the first of its kind, in a sense the most essential question that I am attempting to answer here is this:

> **What is a computational media dissertation?**

I hope that I have done well to answer this question. As you journey through this tome, godspeed!

Part I

Why and How

Chapter 2

Defining Emergent Narrative

This dissertation is about emergent narrative: its pleasure, its pain, a framework for harnessing the former while simultaneously rectifying the latter, and a retrospective of projects that demonstrate that framework. Since it is, as such, our center of gravity, it would seem prudent to first explicitly define the term 'emergent narrative' (as I treat it in this document). In this chapter, I develop a definition for the concept, explore the origins of the term 'emergent narrative', and then conclude with a brief overview of academic work in this area. In the latter section, I give particular emphasis to the academic work of Ruth Aylett, Sandy Louchart, Mariët Theune, Ivo Swartjes, and their respective collaborators, since among scholars these research programs are particularly associated with the phrase 'emergent narrative'.

2.1 A Definition

At the level of intuition, 'emergent narrative' means narrative that emerges out of the interaction of processes in a system. In some sense, however, all narrative satisfies this definition. For example, if we were together in person and I were tell you a story, the string of sounds constituting the telling would emerge from the physical articulatory system that enables me to produce speech. At a higher level, the structure of that story would emerge from an even more complex system: my brain, or perhaps a larger system in which my brain is embedded, such as my body or my being or the world or the universe, or the alpha and the omega and the interstices, et cetera, and so forth.[1] By such an inclusive definition, all narrative might be considered emergent, but clearly this is too broad for the purposes of this investigation.

The Importance of Computation

Let us say, then, that in this document 'emergent narrative' refers to narrative that emerges out of the interaction of processes in a *digital* system. Still, this is too inclusive: when a work of narrative cinema is played on a DVD player, the sequence of frames that compose it are in a sense emerging out of the digital system constituted on the player, but that is not what I mean by 'emergent narrative'. How about we swap out 'digital' for 'computational'? This is better, but even DVD players use computation, and movies that use computer-generated imagery do so through the utilization of computational systems that enable the authoring and rendering of such imagery (at production time). More problematically, all videogame narrative seems to fall under this definition—even cutscenes require some computation, let alone configurations that incorporate player interactivity.

So, what is missing so far? How is emergent narrative different than other

[1]Richard Walsh has essentially made this argument in describing narrative creation as a complex, dynamical process out of which artifacts emerge [1307, pp. 130–147]. Others have produced similar arguments with regard to reader cognition—due to the intersubjective nature of storytelling, a story *as mentally processed by a given interlocutor* may be viewed as a kind of idiosyncratic emergent narrative [68, 276, 407]. To quote Elinor Ochs and Lisa Capps: "Narratives are tales that tellers and listeners map onto tellings of personal experience. In this sense, even the most silent of listeners is an author of an emergent narrative" [890, p. 21].

methods for composing procedural narrative, such as the kind of hand-authored (yet still procedural) approach that is typical of narrative crafting in videogames? While an authored videogame narrative may require the interaction of computational processes to be *presented*, the very constitution of the narrative—comprising its characters, plot, the telling, and so forth—does not emerge from such a system. Emergent narrative, however, does not just require a computational system to be presented, but moreover requires a computational system to be *composed*.[2] In this way, the process of composing emergent narrative is fundamentally removed from human authorship, or more precisely, it is the result of a different mode of human authorship: rather than crafting a narrative artifact, a human crafts a system out of which narrative artifacts may emerge.[3]

Requirement: A Simulated World

We are now very close to the definition employed in this document, but still we are thinking too broadly. To hone further, let us consider a system called

[2]Game designer Marc "MAHK" LeBlanc has made a distinction between emergent narrative and what he terms 'embedded narrative' [660], which is characterized by preconstructed storylines that may be embedded in a gameplay experience (and enacted through gameplay). Later work by Katie Salen and Eric Zimmerman [1079], Henry Jenkins [541], and Ernest W. Adams [12] revisited and expanded upon this notion.

[3]As committee member Ian Horswill noted in feedback on an earlier draft of this thesis, some writers have described the ideation process in fiction writing as working by a kind of mental simulation. For instance, Lajos Egri has expounded an approach to dramatic composition whereby plot emerges from the more primary matter of character behavior [303]. While the metaphor may be compelling, and the craft advice handy, this phenomenon is quite distinct from that of building a computer simulation out of which narrative may emerge. In the latter, character activity is concretized (reified in computational material) and the authoring procedure is more latent—instead of composing a story, one crafts a system that models storyworlds. In this sense, developing a character simulation works similarly to the writing method called *worldbuilding* [1021, 911]. In Section 3.1.3, I will discuss this connection at length. Noah Wardrip-Fruin, in his own feedback on my earlier draft, asked, "Do you want your definition of emergent narrative to require a digital system, or a computational one? For example, if the processes were implemented as a board game, should that fall within the definition?" (personal communication, July 1, 2018). While it could certainly be fruitful to frame the concept of emergent narrative in terms of board games—Thijs Alofs has approached this with the *Interactive Storyteller* project [31, 30, 32]—that would take us outside the purview of this thesis, as the next section clarifies.

Say Anything [1226, 1227], which was developed by Reid Swanson as part of his doctoral work under the tutelage of Andrew S. Gordon. In *Say Anything*, a human interactor and an AI system collaborate to craft a story together by a novel interaction pattern. To start, the interactor provides the first sentence of the story to the system by typing it in free text. Next, the system processes this sentence in an attempt to find the most related sentence to it that appears in a corpus of one million narrative blog posts compiled by Gordon and Swanson [408]. After isolating the most related sentence in the blog corpus, the system retrieves the sentence that immediately *succeeds* that sentence in the blog post in which it appeared. The idea here is that the successor sentence in the original blog post may serve as a good response to what the interactor provided, since a human writer decided to place the successor sentence after a sentence that is related to what the interactor provided.[4] After retrieving the successor sentence, the system adapts it to better match the prompt provided by the interactor (e.g., by replacing character names), and then appends it to the unfolding story. From here, the interactor provides another sentence, the system forms a response to that next sentence, and so forth until the story has concluded.

While narrative emerges bottom-up in *Say Anything* through the interaction of processes in a system that is partly computational, it is not an example of emergent narrative as the notion is typically conceived. This is due primarily to the absence in this system of a fundamental component of emergent-narrative systems: a *simulated storyworld*. In emergent narrative, a computer simulation models a storyworld, and narrative emerges as the byproduct of running the simulation over a series of timesteps.[5] Typically, or perhaps universally, the storyworld features

[4]In artificial intelligence, this is called *case-based reasoning* [612].

[5]In his feedback on an earlier draft of this thesis, committee member Jonathan Lessard noted that I do not define or even discuss my usage of the term 'simulation', which is curious given its centrality to the concerns herein. My lack of consideration in this regard is in line with

characters who to some degree act autonomously, in which case the emergent narrative may be more concretely conceived as being constituted in the accumulation of character activity over the course of simulated story time.[6] If the system is made interactive—for instance, by allowing an interactor to decide how one or more of the characters will act—then the interactor is to some degree responsible for the narrative that emerges over the course of her interaction with the system. While this is also the case with *Say Anything*, such a configuration resides more firmly in the emergent-narrative tradition because the resulting narrative

conventional discourse in the academic area of videogames, as Jonathan noted in pointing me toward a fantastic paper by Veli-Matti Karhulahti titled "Do Videogames Simulate? Virtuality and Imitation in the Philosophy of Simulation" [567]. Karhulahti explains that, while participants in the *ludology vs. narratology debate* [356, 869, 767] took one another to task vis-à-vis the appropriation of narratological concepts in the nascent scholarly area of game studies, statements such as "All computer games are simulations" [916, p. 1] have been curiously uncontested. Writing about discourses on the philosophy of simulation and in simulation research—where, for instance, the terms 'computer simulation' and 'computerized simulation' have distinct meanings [567, p. 3]—Karhulahti argues that the essential prerequisite for simulation inasmuch as those fields are concerned, that a reference system is imitated, is generally violated in the case of videogame "simulations". While he concedes that a peculiar, but now conventional, sense of the term seems to have evolved in computational media, Karhulahti worries that this challenges cross-scientific communication: "I recommend videogame scholars not to pass the word simulation through duty free interdisciplinarity, but to reserve the word simulation for those already-numerous, well-established purposes that have been recognized above" [567, p. 12]. Instead, he recommends the term 'virtualization' (and, in turn, 'virtual universe') for cases where no reference system is modeled. In this dissertation, and in my life, I am guilty of Karhulahti's charges. The historically conventional sense of the term—by which I mean something like 'an artificial system that imitates a natural one'—strikes me as curiously narrow. My introduction to the phenomenon of simulation was through videogames as a small child, so it never occurred to me that a simulation should imitate some reference system. To me, the term suggests an artifact that is synthetic, processual, and has the quality of being its own enclosed world. In saying this, I am rearticulating the unexamined sense of the term that Karhulahti attributes to videogame scholars, but what I mean to express is that I personally developed this (mis)understanding through my experience in the world: in my exposure to the term, 'simulation' has been almost exclusively associated with videogames. When the world changes, words change, and I personally do not feel beholden to an etymologic or scientific basis for the "true" meaning of 'simulation'. While I am thoroughly impressed by Karhulahti's argument—his paper is fabulous—I will continue to employ the (arguably) degenerate sense of the term.

[6] I can imagine an emergent-narrative system that would not model characters, but it would simply be one whose stories are not about characters. Still, in order to produce narrative, the system would have to model *something* that could serve as the subject(s) of its emergent stories, and I think the term 'character' is a close enough approximation to 'subject', and it is certainly a more natural general term.

emerges not in a general sense, but more specifically *emerges out of a character simulation*—this is the defining quality of emergent narrative.

Critically, the degree to which a system is situated in the emergent-narrative paradigm depends on what might be referred to as the *integrity* of its character simulation.[7] In the area of *narrative planning* [1370, 1004], for instance, systems commonly modulate character behavior (or other simulated phenomena) according to the activity of a plot-level agent that is removed from the storyworld in which the characters are diegetically situated. When the simulation is tampered with in this way, stories no longer emerge, but instead spawn according to the policies of the interventionist mechanism—in the wake of such tactics, a work of emergent narrative is transformed into something else entirely.[8] Here, I mean to speak primarily in aesthetic terms (as cued by the value-laden language of 'tampering' with a simulation), since emergent narrative is really just an artistic genre. While the form is associated with a set of architectural features and technical strategies, I view these as ultimately constituting a poetics that is in the service of a specific pleasure: emergent stories are a joy to behold because it feels like they actually happen. In the next chapter, I unpack this idea further.

Converging on a Definition

Let us now converge on a concise definition that captures all the features outlined so far. First, though, I have a final subtlety to note: 'emergent narrative' refers to the approach to procedural narrative that I have described in

[7]Specifically, what someone *with my artistic inclination* might refer to as 'integrity'. Again, this document is in part a manifesto.

[8]Generally, simulation integrity is compromised in an effort to guarantee story structure. While such tactics have been demonstrated to work well, they undermine the distinct pleasures of emergent narrative, which I discuss in the next chapter. This dissertation revolves around a novel approach to emergent narrative that guarantees story structure without intervening in the storyworld, thereby maintaining the pleasures of the form.

this section, but it can also refer to actual narrative material generated by a system taking that approach. So, for the purposes of this document, our definition of 'emergent narrative' is as follows: *narrative that emerges out of computer simulation of character activity, or the methodology of generating narrative in that way.* Critically, this definition does not stipulate that a work of emergent narrative must be interactive. While academic work in the area has typically treated emergent narrative as (specifically) an approach to interactive storytelling [50, 54, 701, 702, 1233, 1230, 1308, 1319, 1272], I take a more ecumenical stance by which three distinct traditions are treated as instantiating the same essential form: the simulationist school of noninteractive story generation, as exemplified by systems like *Saga II* [859], *Tale-Spin* [822], and Sheldon Klein's murder-mystery generator [601, 597]; videogames driven by emergent character simulation, such as *Dwarf Fortress* [17] and *The Sims* [792]; and the bottom-up approach to academic interactive storytelling that is exemplified by the 'emergent narrative' research programs of Ruth Aylett, Sandy Louchart, Mariët Theune, Ivo Swartjes, and their respective collaborators.

the methodology characterized by computational systems in which narrative emerges bottom-up from the interaction of processes in underlying simulations that typically feature autonomous characters (or, alternatively, the actual narrative material produced by this method).

'Narrative' Confusion

Note, finally, that while I have taken great pains to define the particular meaning of 'emergent' that is operational in the term 'emergent narrative', I have not spent any effort on 'narrative'. This is because the issue of what exactly 'narrative' means is murky, particularly in procedural narrative [505, pp. 82–83], and especially in emergent narrative, as I will discuss at length later on. First, at a general level, the word itself has multiple conflicting usages: ‘narrative’ can be used in the singular to denote a concrete story (and in the plural to denote multiple stories), or it can be used as a mass noun to denote the abstract notion of narrative itself. Moreover, the words ‘narrative’ and ‘story’ are sometimes used as terms of art that correspond to the notions of *fabula* (the sequence of events recounted in a story) and *syuzhet* (a story as it is told), respectively. More confusingly still, these particular usages of ‘narrative’ and ‘story’ are sometimes eschewed in favor of the respective alternatives ‘story’ and ‘discourse’ [177]. This terminological variation is couched in a century-long tradition of structuralist work in narratology, as Michael Scheffel recounts concisely (citations inserted for clarity):

> Various binary oppositions have been put forward, such as *fabula/sujet* (e.g. Tomaševskij 1925 [1262]), *histoire/discours* (e.g. Todorov 1966 [1258]; story/discourse), and story/plot (e.g. Forster 1927 [350]), as have multileveled models such as *Geschehen/Geschichte/Text der Geschichte* (Stierle 1971 [1204]; happenings/story/text of the story), *histoire/récit/narration* (Genette 1972 [384], 1983 [386]; story/narrative/narrating), and *Geschehen/Geschichte/Erzählung/Präsentation der Erzählung* (Schmid 1982 [1113]; happenings/story/narration/presentation of the narration). [1108, para. 4][9]

Let me emphasize this situation: some scholars use ‘story’ to mean ‘fabula’, while others use it to mean the counterpart, ‘syuzhet’—thus, with regard to this binary classification, ‘story’ can mean one thing or the complete opposite.[10] For this reason, and because ‘story’ and ‘narrative’ already have extensive informal usage, I prefer to use the terms ‘fabula’ and ‘syuzhet’ in cases where the formal narratological concepts need to be referenced. In composing this document, however, I have not found these concepts to be of much use, primarily for the reason

[9]See Scheffel’s full paper for an excellent brief history of this narratological tradition [1108].

[10]The issue is here has certainly been muddled by challenges rooted in translation. For more on this confusion, see Richard Walsh’s account [1307, pp. 53–56].

that they do not apply particularly well to the case of emergent narrative.

A Note on Terminology

As I will argue at length in Section 3.1.1, emergent narrative actually works more like nonfiction than fiction, which means that formalisms that have been developed specifically for the latter—such as the distinction between fabula and syuzhet—do not apply particularly well to this domain. In fiction writing, we may, for instance, distinguish between the order of story events as they transpired (in a storyworld), as opposed to the order in which they are recounted. While this is also true of narrative nonfiction, in that genre there is the even more primary matter of all the occurring events (in the real world) that are not recounted or even suggested in the surface account—in fiction, this base layer cannot obtain, because events that are not even suggested cannot be known to have occurred.

As such, we find that something like a three-tiered typology serves better: all the events of a world, the events that are recounted, and those events *as* recounted. Indeed, this is the essence of some of the models included in the quotation of Scheffel above, most famously that of Gérard Genette. This is the one preferred by Jonathan Lessard, a member of the reading committee for this thesis, who suggested, in his comments on an earlier draft, that I consistently use Genette's terminology (specifically, the common translations 'narrative' and 'story' to refer to the latter two concepts that Genette called 'récit' and 'narration'). This is certainly good advice, but as I mentioned above, I do not like to use 'story' and 'narrative' formally, on account of their well-established informal senses that denote roughly equivalent concepts. As such, in this document, I use these terms informally (and interchangeably) to mean something like 'concrete narrative artifact'. Additionally, I will employ phrases like 'plot' or 'raw narrative material' to refer to specific senses of ‘fabula’, and phrases like ‘presentation’, ‘telling’, or ‘surface text’ to refer to specific senses of ‘syuzhet’.

While in those cases I choose my words informally, in this thesis I *do* adopt one particular typology whose terms I use consistently: Hayden White’s model of historiography [1332], and in turn, the terms ‘historical field’, ‘chronicle’, ‘story’, and ‘emplotment’. Indeed, it is in White’s model, as opposed to any narratological formalism targeted at fiction, that I find the strongest application to emergent narrative. This is because I believe that the latter works more like nonfiction than fiction—as I have already stated and will argue at length below—which makes the task of crafting emergent narrative strikingly akin to the practice of history writing. In the next chapter, I dive thoroughly into these concerns, but for now I would to emphasize again that in this document I do not formally employ any of the narratological terms of art quoted above (because I do not think that they apply particularly well to the concerns of emergent narrative).

Preview: Narrativism

Finally, to add to the terminological confusion that I have just outlined, the shape of the narrative artifact produced in emergent narrative is often nebulous. As I will explain at length in Chapter 4, in many projects the total history of everything that has been simulated is treated as the emergent narrative—but this is peculiar, because such accumulations of simulated material lack the features that are associated with narratological or informal definitions of story. Even stories that recount lived experience have far more narrative structure. By analogy, if such stories worked like emergent narrative tends to work, then the lived experience *itself* would be the narrative, rather than a story constructed to impart the lived experience. But few would call lived experience itself a story: no one says

‘You won’t believe the story that happened to me today.’[11]

This confabulation is what scholar Espen Aarseth has termed ‘narrativism’ [4]. As I discuss more thoroughly in Chapter 4 (in extending critiques by Aarseth and others), I find the treatment of accumulated simulated material as narrative to be problematic, and I think it is at the root of some of emergent narrative’s failings to date. The core subject of this dissertation, as Chapter 5 hammers home, is a refined approach to emergent narrative that entails the *curation* of raw simulated material into an actual narrative artifact, which is itself mounted in a full-fledged media experience. Again, I call this *curationist emergent narrative.*

2.2 Origins of the Term

In this section, I will briefly outline the provenance of the term ‘emergent narrative’. To my knowledge, this history has not been discussed (to this extent) in any prior publication.

Precursor Terms

The actual approach denoted by the term ‘emergent narrative’ precedes its 1990s coinage (about which more soon) by decades. As I explain in Section 4.1.2, the earliest known effort in story generation, *Saga II* (1960) [859], is a work of emergent narrative: in this project, stories about Wild West altercations are generated by simulating the activity of two autonomous characters. Other more famous early systems, namely Sheldon Klein’s murder-mystery generator (1967–1973) [601, 597] and Jim Meehan’s *Tale-Spin* (1975–1977) [822], are also associated

[11] Instead, one says something like ‘You won’t believe the story *of* what happened to me today.’ Here, ‘of’ is critical: the story is constructed *out of* the raw material of lived experience, which is not in and of itself a story. To claim otherwise is to conflate experience with narration, and in emergent narrative we find a related misconception: simulation is confused with narration.

with the approach.[12] Thus, we may look at how these projects were described to identify precursor terms such as "world simulation based story generation" [252, p. 16] or "the simulative approach" to story generation [1065, p. 513].

Environmental Storytelling as 'Emergent Narrative'

While the collocation 'emergent narrative' appears incidentally (with alternative connotations) in prior works,[13] the earliest usage that I have found with reference to procedural narrative appeared in *The Observer* in November 1994. In an article about the advent of "interactive movie games" (videogames that integrate full-motion video footage [678]), Jim McClellan reviews the newly released Philips CD-i game *Burn:Cycle* [1270]. As McClellan notes, new mediums (such as interactive movies) tend to beget new turns of phrases: "I suppose when you're scratching together a new hybrid media form, you get to not only write your own rules but come up with your own terminology" [795, p. 84]. One of the new phrases coined by *Burn:Cycle* developers Dave Collier and Olaf Wendt, who McClellan interviewed for the piece, was 'emergent narrative':

> The idea is to explore, pick up clues and uncover the narrative that has been plotted out in advance. "I think what we're doing is giving people a lot of movie components, like beads, and saying: 'You put the thread through,'" says Collier. "It's what we call emergent narrative." [795, p. 84]

[12]Both of these systems are discussed at length later on. These are not the date ranges that have been associated with those two systems in our field, but they are attested in documents that I have recently excavated. In a 1969 note on his "automatic novel writing" project, Sheldon Klein lists a project start date of September 1967 [590, p. 418]. Likewise, while *Tale-Spin*'s first reporting is conventionally understood to have been in Meehan's 1976 thesis [822], by then he had already published a now obscure article on the project [821] in a 1975 issue of the *American Journal of Computational Linguistics*.

[13]One such paper even has the tantalizing title "Interactive Storytelling", but it uses that phrase in the sense of interactive conversational storytelling, a practice that "provides a helpful structure for a young child's emergent narrative ability" [1271, p. 169]. Another paper actually describes a kind of emergent narrative constituted in the dynamics of an annual car race and its host city [922], but this is still not what we mean by the term here.

Since interactive movies represent arguably the antithesis of emergent narrative, at least in terms of procedural approaches, it is not surprising that this connotation does not coincide with the contemporary sense of the term. Interestingly, however, Collier and Wendt's 'emergent narrative' seems to almost perfectly denote what today would be called *environmental storytelling* [164, 165, 924].[14]

Galyean's Forgotten Coinage

When it comes to the current sense of 'emergent narrative', we find a first usage in the work of Tinsley "Tag" Galyean, who at the time was a PhD student at MIT. In a 1995 paper presented at the AAAI Spring Symposium on Interactive Story Systems: Plot and Story,[15] Galyean articulates an interventionist approach

[14]A concept borrowed from theme-park design, *environmental storytelling* is a technique in videogame design by which a game's narrative is relayed primarily through the mechanism of its spatial enviroment (and often a series artifacts that are distributed across that environment). I prefer Clara Fernández-Vara's refinement, *indexical storytelling*, which more deeply considers the role of the player: "indexical storytelling is actually more story-building, both on the part of the designer and the player—the designer creates the elements of the story and integrates them in the world, the player has to interpret them and piece them together" [336, p. 6]. Moreover, Fernández-Vara's framework also considers how a player may leave such traces in a world, thereby inscribing her own story onto it, as another kind of indexical storytelling. Though conventional environmental storytelling, such as that found in so-called *walking simulators* [158], may be antithetical to emergent narrative, I see indexical storytelling as actually being quite compatible. For one, world simulations are apt at incorporating player behavior, which makes player-driven indexical storytelling a strong prospect in works of emergent narrative. I am also interested in the prospect of *generative* environmental storytelling. As I note at several points later on, this is arguably an unavoidable feature of world generation, especially when artifacts are modeled: the result of such a procedure is a storyworld scattered with indices that encode and suggest the history of that world. What I would personally like to see is exploration of game designs that are specifically built on this idea. In a recent paper, Daniel Livingstone and collaborators have promoted this phenomenon in terms of what they call *archaeological storytelling* [694]. Relatedly, scholars in the larger area of *archaeogaming*, which is about "archaeology in and of videogames" [994, 993], promote an approach to gameplay that treats simulated storyworlds as archaeological sites (for instance, the worlds of *No Man's Sky* [345]). More broadly, Tanya X. Short has called for simulationist game designs that encourage player investigation of simulated material [1150, p. 114-115].

[15]This symposium was an important early meeting in the academic field of interactive storytelling, which is today represented primarily by the workshop on Intelligent Narrative Technologies and the International Conference on Interactive Digital Storytelling (ICIDS). These were preceded by the AAAI Fall Symposium on Narrative Intelligence (1999), as well as the International Conference on Virtual Storytelling (2001–2007) and the International Conference on

to interactive storytelling akin to drama management, which he calls "narrative guidance" [370].[16] Critically, Galyean positions this approach in opposition to what he calls 'emergent narrative', defined as follows:

> We all construct narratives out of our daily activities to help us remember, understand, categorize and share experiences. It is this skill that many interactive systems exploit. They give us environments to explore. We, by combining the elements of these spaces with our goals (the user's goals), allow a narrative to emerge. If any narrative structure (or story) emerges it is a product of our interactions and goals as we navigate the experience. I call this "Emergent Narrative." This approach has provided a number of successful interactive experiences such as flight simulators, games (i.e. DOOM), and narrative puzzles like MIST [Myst]. [370, p. 1]

While this definition differs from mine, it is quite a bit closer than the sense of the phrase discussed in the previous section. In a paper presented a few weeks later at another symposium, Galyean describes the notion again:

> Architectural walk throughs for example, scientific visualization, and even games like DOOM place us in alternative worlds while giving us methods for navigating these virtual spaces. These methods allow smooth and continuous interaction that can immediately influence the

Technologies for Interactive Digital Storytelling and Entertainment (2003–2006), which merged to form ICIDS. Sadly, while each of the other eight AAAI Spring Symposia of 1995 published proceedings, Interactive Story Systems: Plot and Story did not (I confirmed this with AAAI). However, it appears that working notes for the symposium (comprising the papers) were distributed to attendees, so there is hope that this (otherwise lost) pioneering historical work is extant. Among its papers, for example, is one by Janet Murray apparently titled "Dr. Quinn on the Holodeck, or blueprint for an electronic storyland" [868] (as cited later by Clark Elliot [308, p. 188])—this would have preceded her famous *Hamlet on the Holodeck* [870] by two years. To my knowledge, however, the *first* academic meeting in the area was the 1990 AAAI Workshop on Interactive Fiction and Synthetic Realities, which was organized by Joe Bates and Abbe Don (and for which a proceedings was likewise not published) [81]. In a 1991 paper, Bates stated that the workshop was at that point "the only place that workers in the relevant fields have come together to seriously discuss broad research toward inhabited, dramatic virtual worlds" [84, p. 3]. Elsewhere, in a newsgroup post, Bates explained that the field was "starting to become acceptable as a 'serious' AI research area" [82, n.p.]. Among researchers writing in this century, only Federico Peinado has referenced this meeting [925, p. 43].

[16] Curiously, this paper does not cite anything else, let alone the highly related work on drama management carried out by the symposium's co-organizer, Joe Bates [575]. To be fair, Galyean takes a distinctly cinematic perspective and is particularly interested in how, for example, camera control may be used as a technique for dramatic intervention, as his dissertation elaborates [371].

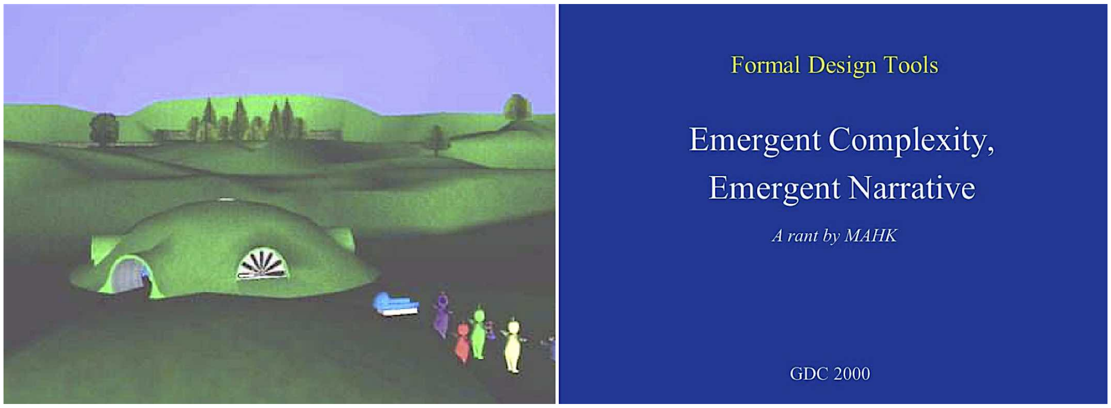


Figure 2.1: Images associated with the invention of the term ‘emergent narrative’ in the 1990s. Though it had appeared in earlier contexts, namely in a series of 1995 papers by Tinsley Galyean, the term ‘emergent narrative’ was popularized in academia through its use by Ruth Aylett with regard to her *Virtual Teletubbies* project of 1999 (left). Around the same time, Marc LeBlanc introduced the term into the parlance of game design through a seminal presentation at the 2000 Game Developers Conference (right). It appears that these three coinages occurred independently of one another, which suggests that the phrase was a natural construction by the mid-1990s.

> constantly changing presentation, but they rely on the user’s actions and thoughts to bring structure to the experience. If any narrative structure (or story) emerges it is a product of our interactions and goals as we navigate the experience. I call this emergent narrative. [369, p. 103]

Aylett’s Academic Usage

Galyean’s term did not catch on, perhaps because his contritubitions that introduced the idea were actually about an oppositional interventionist approach (his ‘narrative guidance’). In any event, it was a few years later, at the 1999 AAAI Fall Symposium on Narrative Intelligence [772], that the phrase ‘emergent narrative’ would come into wider usage in the field. Here, Ruth Aylett presented a (now seminal) paper titled “Narrative in Virtual Environments—Towards Emergent Narrative”, which introduced the bottom-up character-centric approach that she had been exploring in her (yet unpublished) *Virtual Teletubbies* project [54] (see Figure 2.1). Aylett does not explicitly define 'emergent narrative' in the paper, but instead treats it as an intuitive concept that can be explained as the bottom-up mechanism by which narrative obtains in example domains like soccer matches and improv scenes [50, pp. 84–85].[17] From here, Aylett's research program took off, producing a series of papers on the topic (discussed in the next section), and the term 'emergent narrative' became a common token in the academic field of interactive storytelling (though one sometimes specifically associated with Aylett's program).

LeBlanc's Game-Design Usage

Around this same time, the term 'emergent narrative' also entered the parlance of game design, both in academia and industry, through the apparently independent coinage of game designer Marc "MAHK" LeBlanc. At the Game Developers Conference in 2000, LeBlanc delivered a presentation titled "Formal Design Tools: Emergent Complexity, Emergent Narrative". Video does not appear to survive (though LeBlanc's slide deck is available online [660]), so I will

[17]In several earlier papers by Barbara Hayes-Roth and collaborators (beginning in 1995), the patterns of improv acting are invoked to articulate a new paradigm for procedural narrative [463, 462, 465, 461, 467, 468]. Ken Perlin and Athomas Goldberg also adopted this perspective in an antecedent project [939]. However, none of these publications use the language of 'emergence' to describe their systems. A striking 1994 article by John Barger (the inventor of the term 'blog') speaks of authoring toward "the stories you want to see 'emerge'" in a blue-sky system for interactive storytelling [74, n.p]. With uncanny prescience, Barger almost exactly describes *The Sims*, even using the moniker "SimNormal": "I like to imagine a sort of SimCity-like environment where a community of little NPC's go through their daily routines of life, in a completely rote fashion [...] In honor of Normal, Illinois, I call this world "SimNormal". [...] So the game might open with a person in their apartment, waking up in the morning, and if they go outdoors they may be presented with a series of neighbors whose dramas they can become involved in [...] If the player chooses not to follow up any given opening, that's no problem... there will be many other options to explore" [74, n.p.]. Alternatively, a fascinating 1998 paper by Peter Bøgh Andersen considers the physical sense of 'emergence' in describing *The Eye of Wodon*, whose storyworld is represented as a dynamical system that both the player and the system perturb [38]. For a deep dive into the connection between dynamical systems and narrative, see Jo Alyson Parker's monograph *Narrative Form and Chaos Theory in Sterne, Proust, Woolf, and Faulker* [915].

quote from Ernest W. Adams's summary of the talk (included in a 2005 lecture):

> At the 2000 Game Developers' Conference, Marc LeBlanc gave a lecture called "Emergent Complexity, Emergent Narrative." He introduced the idea that narrative can emerge from complex automated systems rather than from pre-written blocks of material. He made a distinction between what he called "embedded" and "emergent" narrative. Embedded narrative is pre-constructed, and the player encounters and experiences it in the course of gameplay. Emergent narrative arises out of the process of playing. [12, n.p.]

As Adams notes, LeBlanc's ideas were not new—in a 1995 lecture on interactive movies, Adams himself commanded: "Your job is not to tell stories; your job is to build worlds in which stories can happen" [11, n.p.]—but nonetheless his explicit formalization, and especially the concise term 'emergent narrative', were helpful additions to the game designer's conceptual library (and lexicon).[18]

A Commonplace Term

Shortly after LeBlanc's presentation, game designer Eric Zimmerman referenced his concept of emergent narrative in a piece titled "Against Hypertext" (2000) [1379]. In 2004, Zimmerman co-authored (along with Katie Salen) a popular book on game design, *Rules of Play: Game Design Fundamentals*, which includes several segments on emergent narrative [1079, pp. 383–385, 402, 418, 579–580]. As the authors acknowledge, they adapt the concept from LeBlanc's 2000 presentation,[19] and from here the term (and concrete idea) became firmly entrenched in the (conceptual) lexicon of game design. Indeed, it is likely primarily

[18]Confusingly, LeBlanc's other coinage, 'embedded narrative', coincides with Marie-Laure Ryan's term for "any story-like representation produced in the mind of a character and reproduced in the mind of a reader" [1063, p. 320]. As Ryan acknowledges, the term has also been used more loosely by narratologists to refer to any narrative material embedded in other narrative material, such as in the case of a frame story [69].

[19]Though Salen and Zimmerman accidentally cite LeBlanc's Game Developers Conference talk of 1999, which described feedback systems in games (and was built upon by his 2000 presentation on emergent narrative) [659].

through this book that the term has found widespread usage in lay circles.[20]

Finally, it is worth noting two other early usages of the term in influential papers, both of which appeared in the 2004 volume *First Person: New Media As Story, Performance, and Game*, edited by Pat Harrigan and Noah-Wardrip-Fruin. In the seminal "Game Design as Narrative Architecture" [541], Henry Jenkins utilizes the terms 'embedded narrative' and 'emergent narrative' with LeBlanc's same senses for them, but does so without citation (nor is Aylett or Galyean cited). A few pages later, in "Towards a Game Theory of Game" [923], Celia Pearce uses 'emergent narrative' as a commonplace term (i.e., undefined, without citation).[21] Interestingly, these scholars would likely have been exposed to both Aylett's and LeBlanc's ideas. In any event, it is safe to say that the term 'emergent narrative' was in common use—"in the zeitgeist" [129]—by 2004.

2.3 Brief Overview

As I noted in the last section, the emergentist approach to procedural narrative is as old as the field itself, originating in some of the earliest known story generators: *Saga II*, *Tale-Spin*, and Sheldon Klein's automatic novel writer. Many more projects have followed in recent decades, including academic systems that demonstrate AI research, like *Virtual Storyteller* [1246], *FearNot!* [55], and *Prom Week* [803], as well as commercial games like *The Sims* [792], *Dwarf Fortress* [17], *RimWorld* [1209], and *Caves of Qud* [428]. Other character-based systems, such as the *Petz* series of digital games [352], the interactive-story system of Marc Cavazza and others [170], and Chris Crawford's *Storytron* [224], could also be

[20]To be clear, others were also writing at this time about the more general phenomenon of emergence in games. Here, a seminal example is Jesper Juul's 2002 paper "The Open and the Closed: Games of Emergence and Games of Progression" [556].

[21]In his book *Half-Real*, published the following year, Jesper Juul critiques the concept by referencing both Jenkins's and Pearce's articles as its primary articulations [557, pp. 157–159].

called works of emergent narrative.

The *emergent narrative* research programs cultivated by Ruth Aylett, Sandy Louchart, Mariët Theune, Ivo Swartjes, and others have typically concerned only interactive systems, most often research systems like *FearNot!* [55] and *Virtual Storyteller* [1246]. As I have stated already, in this dissertation, I do not limit my concerns to interactive systems. A basic appeal of emergent narrative is of course the very *emergence* that it yields, producing stories that even a system's designer might not have anticipated. Indeed, games with rich underlying simulations have often been lauded for their emergent qualities [556, 23, 1334]—this is what got *Dwarf Fortress* into the Museum of Modern Art [1344]. The fundamental rhetoric in academic work advocating this bottom-up approach is that it defeats the sticky issue in top-down interactive narrative of accommodating player actions [1003, 50]. If the affordances given to a player are a subset of (or are coextensive with) the actions that non-player characters (NPCs) may themselves take in an underlying simulation, then an emergent-narrative system will be inherently reactive to player inputs. As such, the stories that emerge in these systems naturally incorporate player actions. In Aylett's early formulation, emergent narrative was articulated as a solution to this problem of conciliating interactivity and narrativity [50], which has been called the *narrative paradox* [702, 55] and the *interaction dilemma* [926]. I am not aware of any championing of the form outside of the context of interactive experiences, though several noninteractive story generators have taken the approach. Generally, the simulationist approach to story generation has been lambasted, as I will explain at length in Chapter 4.

Papers in the area of emergent narrative have typically introduced new systems [55, 142, 34] or more generally concerned the engineering of systems [701, 53, 1319, 1272, 107], though others have discussed emergent narrative through the lens of narrative theory [703, 702], creativity [617, 977, 52], and improvisation [1234, 1230]. Finally, another class of contributions to the discipline has concerned the particular *design practices* that are employed in the domain [705, 704, 1223, 618, 1087].

In this dissertation, I provide what is to my knowledge the most extensive *apologetics* for emergent narrative, with particular attention to the hidden merits of dismissed noninteractive systems, such as Sheldon Klein's murder-mystery generator [601, 597]. In turn, I also provide what I believe to be the most extensive *polemic* for the form to have appeared so far. Here, my approach is to paradoxically take the most devastating takedowns of emergent narrative as an intellectual *starting point* for a refined instantiation of the form, *curationist emergent narrative.* Throughout the discussion, I will weave intellectual, technical, design, and historical threads into a holistic account of what has happened so far and where we might go next.

Chapter 3

The Pleasure of Emergent Narrative

Through its utilization in important videogames such as *Dwarf Fortress* and *The Sims*, emergent narrative represents perhaps the most successful approach to procedural narrative in the public consciousness. But why are these works beloved—why do people like them? What is the pleasure of emergent narrative? In this chapter, I attempt to answer this question by discussing a series of related phenomena with which emergent narrative shares critical features: nonfiction, stories of lived experience, worldbuilding, and art brut. Finally, I identify a set of aesthetics that are yielded by these shared features. I call these the *aesthetics of emergent narrative*, and I believe that they undergird the pleasure of the form (and thus provide an account for why people like its successful examples). Throughout the chapter, I refrain from discussing my own work, though in Part II I will apply the intellectual framework developed here in the case studies included there. Nonetheless, I should note that the ideas articulated in this chapter were developed not only through interpretation of the projects discussed herein, but also through the evolution of my own simulation and media practice. As such, my autoethnography in Part II tells a story about how I arrived at these arguments.

3.1 Analogous Forms

In this section I will articulate a series of analogies by which we may view emergent narrative as having certain structural, functional, or superficial resemblances to other narrative and artistic forms: nonfiction, stories of lived experience, worldbuilding, and art brut. With this foundation in place—particularly the resemblances to nonfiction—it becomes feasible to identify a set of aesthetics that are distinctly associated with emergent narrative, which I provide at the end of this chapter.

3.1.1 Analogy to Nonfiction

Emergent narrative works like nonfiction. Here is what I mean: events that emerge out of the interaction of processes in a computational system actually *happen*, and so stories that recount such events work like nonfiction, to the degree that nonfiction is conceived as the recounting of a series of events that have happened. Note that here I mean to call upon the *genre* of nonfiction, as opposed to the *ontology* of nonfiction, the plausibility of which has been challenged by thinkers (both ancient and contemporary) who have argued against the possibility of human objectivity.[1] In this section, I outline an array of scholarly perspectives on nonfiction before providing a case study that applies these perspectives to an example of emergent narrative, to show that it works like nonfiction.[2]

[1]To be clear, by 'genre of nonfiction', I mean narrative works that (ostensibly) faithfully recount a series of (ostensibly) real events, as opposed to, for example, a math textbook or a manual for operating a typewriter. I will elaborate on this notion below.

[2]My argument in this section is essential to the ideas promulgated in this thesis, and so I have sought to be diligent in making it. Starting now, I will depart on a rather adventurous foray into a wide array of scholarship on the ontological status of nonfiction. While some of

Essential Objections to Objectivity

There are ancient arguments that question whether there is even an external world that is constituted in a reality. For example, the solipsistic position holds that only the self can be known to exist [49]. A more extreme position, attributed to the Greek sophist Gorgias, contends that in fact *nothing* exists, and even if something did exist, we could not perceive it, and even if we could, we could not impart such perception [576]. Thus, if nonfiction is the recounting of a sequence of real events, and the very notion of a real event (or realness itself) is called into question, then the ontology of nonfiction, as typically conceived, is challenged. Alternatively, if reality exists but is constituted by oneself, or if an external reality can be perceived but not recounted to others, then the very project of nonfiction is frivolous—it is just the recording of a diary for one's own consumption.

Contemporary Objections to Objectivity

Even if we believe in the possibility of real events existing and being recounted, contemporary scholarship (especially from postmodernism) problematizes the notion of *faithfully* recounting real events. Here, an essential critique maintains that while a true account of reality would be *objective*, humans are subjects whose experiences, understandings, and expressions are (thereby) inherently *subjective* [351, 1023, 520]—Satya Mohanty has called such denial of objectivity "the core thesis of postmodern literary and cultural theory" [842, p. 10]. Due to its predication on human experience, understanding, and expression, the very project of nonfiction is subjective by this view. Moreover, even if the propositional content expressed in a work is somehow all true, the work will (of course) still exclude

---

the upcoming discussion may feel at best tangential, trust me when I say that all of it is in fact highly relevant to the core argument of this section: that emergent narrative works like nonfiction. I promise that the journey will make sense by the destination.

many truths about reality, and so its particular inclusions and exclusions are the byproduct of a subjective process of selection. Indeed, this selection—moreover, any mode of artistic or scientific or generally communicative production—is necessarily couched in relativistic social and cultural practices [621, 100, 1300, 940].

Roland Barthes attacks the idea of nonfiction from a semiotical perspective: the sign system of a recount creates signifieds that may correspond to real referents, but as signifieds they are not literally those referents [78]. He calls the implicit or explicit claims of objectivity in writing a *referential illusion*—the author implies that the signifiers are actually the real referents they signify—that yields a *reality effect* (it produces a style called realism). This referential illusion is typified, Barthes says, by the historian Adolphe Thiers' directive for historiographic practice: "To be simply true, to be what things are and nothing more than that, and nothing except that" [78, p. 148].

Recuperation

So, if nonfiction is not a faithful account of an external reality, is it just fiction? This may be the case in an ontological sense, and that may be unavoidable, but nonetheless there is clearly a thing called 'nonfiction'. Indeed, lay people seem to have little trouble delineating the forms—simply consider how bookstores are organized to feature nonfiction sections.[3] In summarizing the work of Laurence Goldman and Michael Emmison [403, 402], Jean-Marie Schaeffer notes that such

[3]Interestingly, Christine Gallagher argues (partly on the basis of etymology) that the idea of fiction is actually quite new (less than a few centuries old) [368]. Meditating on the origins of language, the writer and thinker E. L. Doctorow has suggested (relaying the opinion of a professor from his college days [1097, p. 124]) the perhaps inevitable case of a time when fiction and nonfiction could not be distinguished because "the designative and evocative functions of language were one and the same" [267, p. 217]. This notion brings to mind the late-career work of Sheldon Klein, a main character in Chapter 4, on the origins of human cognition [598, 593, 594]. Curiously, the word 'nonfiction' seems to express fiction's privileged status as a default mode, as Gérard Genette notes [385, p. 756].

distinction appears to span all human cultures:

> But even if it may be true that fictional narrative as a socially recognized practice is not an interculturally universal fact, all human communities seem to distinguish between actions and discourses that are meant to be taken "seriously" and others whose status is different [1099, p. 99]

Moreover, there are scholars outside the postmodernist tradition who are still interested in characterizing the delineation of fiction and 'nonfiction' (or characterizing, alternatively, a spectrum of fictionality). Such work has tended to appear in three distinct traditions, which may be called the *semantic*, *formalist*, and *pragmatist* perspectives on fictionality. I will discuss the first of these perspectives in the next section (with a particular focus on John Heintz's description of fictional worlds), and the latter two in the subsequent sections.

John Heintz's Fictional Worlds

The *semantic perspective* on nonfiction is characterized by logicians who evaluate the ontological status of narrated entities and the truth values of the propositional content concerning such entities.[4] As Jean-Marie Schaeffer notes [1099, p. 104], this tradition proceeds from the denotation of Gottlob Frege [358] as extended by Bertrand Russell [1035]: a statement's truth value depends on whether that statement obtains with regard to the one actual universe. It is explored primarily in the area of analytic philosophy, where it flourished especially in the 1970s.[5] In an influential paper called "Truth in Fiction" [686], the philosopher David Lewis wrestles with the intuition that a sentence like *Sherlock Holmes lived on Baker Street* seems to be true while *Holmes and Watson are identical* seems

[4] My writing of this section and the following two was greatly aided by Sebastian Deterding, who sent me a helpful array of pointers to scholarly work dissecting the notion of nonfiction.

[5] For more comprehensive reviews, consult the writings of John Woods, who has been active in this area for decades [1355, 1353, 1354]. Matthieu Fontaine and Shahid Rahman also provide a recent overview [346].

to be false.[6] The problem here is that 'Sherlock Holmes' is "denotationless" (it fails to refer to an entity in the actual universe) by Russell's program [686, p. 38]. One potential solution is to take up the Meinongian ontology, whereby any concept that can be described, even a non-existent one, is viewed as an actual object that may be denoted (referred to) [535]. This project has been taken up by Terence Parsons [918], but Lewis cites a number of attendant difficulties [686, p. 37].[7] Instead, Lewis provides a reasonable solution by which (in part) these sentences are taken as implying a prefix, *In such-and-such fiction...*, which means the troublesome sentence is actually understood as something like *In the Sherlock Holmes stories, Sherlock Holmes lived on Baker Street.* Others have posited similar refinements [1352, 918, 473]. By this account, we may say that nonfiction obtains when (or to the degree that) a work's propositions are true without being prefixed in this way.[8] This work, however, is concerned not with the delineation of the fictional and factual, but with building a logic that is robust enough to handle fictional worlds.

This is also the nature of John Heintz's project [473], but his approach is to first identify a set of peculiar logical features that characterize fictional worlds (since a robust logic for such worlds would have to handle such features). Specifically, he argues that fictional worlds are necessarily *incomplete* and are often *inconsistent.*

First, they are incomplete in the sense that they fail the law of bivalence,

[6] John Woods has called sentences like the former *bet-sensitive*: if one person bets another that Holmes lived on Baker Street, the former wins the bet even through both parties will agree that Holmes does not exist [1355].

[7] Parsons's Meinongian view of fictional characters leads to some evocative writing, such as: "I suppose that all of these [fictional characters] were objects before they were written about; they were so to speak only identified by the author, and writing about them did not confer objecthood on them. Nor do I suppose that writing about them brought into existence, for they do not exist" [918, p. 79].

[8] It is interesting to consider this approach with regard to Barthes's referential illusion, mentioned above. Barthes would argue that even if the propositional content of a narrative refers and is all true, the very procedure of denotation makes nonfiction impossible: sign systems cannot contain components of the actual world, but rather only simulacra of those signifieds.

which maintains that, for some proposition p, either p is true or p is false.[9] As Heintz states, for any given fictional world, truth values are undecided for nearly all conceivable propositions about the world:

> authors [of fiction] are not foreign correspondents, reporting, sometimes incompletely, on events they witness in some far-flung corner of the world. They create (most of) the events they write about. What they fail to tell us, either explicitly or by implication, simply does not exist. [473, p. 92][10]

The potential counterargument to Heintz's claim is that the fictional world is an actual one that exists independently of the conception of an author—that is, rather than constructing the world, the author has gained special access to it. This is an interesting idea, but it is not especially convincing, as Heintz argues:

> To believe otherwise is to posit a multiplicity of pre-existing fictional worlds to some of which authors have some kind of special access which, in principle, non-authors might come to share. A decision about the fate of a character could not then be decided by examining the text, [but] by gaining access to the relevant fictional world. We could not just take Barrie's word that Peter Pan could fly; we should be obliged to seek access to the fictional world Barrie was reporting on, and see for ourselves whether or not Peter Pan actually relied on hidden wires or jet propulsion. In the absence of well-articulated arguments for a position with such bizarre consequences, I accept the more natural view that writers make up their stories, and that what they write down is, as far as the fiction goes, the way it is. When nothing in the text supports either [p or $\neg p$], neither is true of that fictional world. [473, p. 92]

Beyond being necessarily incomplete, Heintz argues that fictional worlds are frequently *inconsistent*. As an example, he states that "a careful reading of Anna

[9] Writing later on, Daniel Dennett follows this same reasoning to deem fictional worlds *undecidable* [257].

[10] In the paper, Heintz addresses a clever attempt at producing an exception to this: "Dudley Shapere suggested in conversation that it might be possible to so write a story that every statement about its characters would be decided. For example one might write 'and everything not explicitly stated or implied here is false'. I do not know of any actual cases of this. Of course, if mathematics is any part of the story, then Gödel's incompleteness result, which guarantees that no consistent set of sentences will yield all the truths of arithmetic, rules out even this possibility" [473, p. 91].

Karenina reveals that it begins on a Monday and a Tuesday" [473, p. 92].[11] While such an inconsistency could be viewed as an authorial slip-up that is resolved in the context of the rest of the work, it could of course be possible for such context to variously rely on multiple components of the paradox. If this occurs, the context does not resolve the paradox, but in fact reinforces it. As Heintz puts it, "An author may at one place assert a sentence, at another its denial, and the point of the story turns on both" [473, p. 94].

Let us consider what Heintz's fictional worlds tell us about nonfiction. As noted above, Heintz discusses these characteristics of fictional worlds in pursuit of the development (or identification) of a logic that is robust to such peculiarities. In the case of nonfiction, however, such special logics are not needed, since the world of nonfiction is our external reality: if this is conceived as a sole actual world, Russell's program is robust enough; if instead it is conceived as one of a series of actual possible worlds, then a standard modal logic is robust enough. Thus, we might say that Heintz's description of fictional worlds implies a definition of nonfiction as being constituted in works for which these logics suffice—i.e., works that do not have the troublesome characteristics of fictional worlds that he identifies: incompleteness and inconsistency. Note that this does not necessitate that a nonfiction *narrative* be complete and consistent, but rather that its *subject world* be complete and consistent. Below, I will argue that the subject worlds of emergent narrative (simulated storyworlds) tend to be complete and consistent, and thus the approach itself tends to work like nonfiction.

[11] Apparently this claim was the subject of subsequent discussion that was predicated in part on a misreading of "Monday and Tuesday" as "Monday or Tuesday" [518, pp. 767–768]. For this reason, I quote the passage directly.

Formalist Perspective

As an alternative to semantic approaches, the *formalist perspective* ignores the logical status of narrative content to instead identify the stylistic tropes that characterize works of fiction [1099, 1098, 1100]. For example, Käte Hamburger isolates its means of entering the subjective worlds of characters [445], while Ann Banfield asserts that fiction is marked by discourse that expresses mental perspectives in a third-person voice [72]. Contra these arguments, however, are claims that such features are culturally variable and may be present in both works of fiction and nonfiction, as Henrik Nielson, James Phelan, and Richard Walsh articulate:

> No technique is found in all fiction and/or only in fiction, even though within certain cultural and historical contexts certain textual features can become strong conventional indices of a fictive communicative intent [...] Thus, while some techniques can contextually signal fictive intent, there is no necessary homology of form and function [878, pp. 66–67][12]

Pragmatist Perspective

For scholars taking the *pragmatist perspective*, nonfiction is all about communicative intent: fiction is produced when fiction is intended and fiction is apprehended. This pragmatic view is most associated with John Searle,[13] who views fiction as a string of simulated speech acts: "the author of a work of fiction pretends to perform a series of illocutionary acts" [1122, p. 325]. These speech acts depend on an interlocutor perceiving them correctly, and so Searle speaks also of the "shared pretense" by which pretend speech acts (and their referents) are understood as the author intends [1122, p. 320]. Expanding on this, Stanley Fish

[12]Indeed, as Sebastian Deterding noted in personal correspondence, the genres of *new journalism* [499] and *fake documentary* [661] demonstrate that the stylistic tropes of fiction works may be used in factual accounts and, likewise, the stylistic tropes of factual works may be used in fictional accounts (personal communication, Apr 5, 2018).

[13]Schaeffer asserts [1099, p. 110] that this view actually originates in work as early as David Hume's [519].

notes the activation in fiction (and all communication) of "a set of discourse agreements which are in effect decisions as to what can be stipulated as a fact" [339, p. 1022]. Here, Paul Grice's *cooperative principal* [421] is called to mind [1307, p. 21–23].[14] Of course, a shared pretense might also underpin the construal (by both author and audience) of nonfiction—this has been discussed in the context of myths and sacred texts [1294, 1202]. Scholars beyond the ones I have named have also taken up the project of characterizing fiction according to a notion of speech acts [528, 956, 231]. In a paper preceding Searle's, Richard Ohmann highlights the reader's active role in the shared pretense:

> by providing the reader with impaired and incomplete speech acts which he completes by supplying the appropriate circumstances [...] the literary work is asking him to participate in the imaginative construction of a world—or at least as much of one as is necessary to give the speech acts an adequate setting [893, p. 17]

Along these lines, Dorrit Cohn notes how the fiction writer must do the work of laying *signposts* that call attention to the fictionality of her work [196]. Relatedly, scholars have reported how similar measures may be taken to call attention to the (ostensible) objectivity of a work. In a seminal paper on the craft of newspaper journalism, Gaye Tuchman introduces the notion of *objectivity as a strategic ritual*:

> the correct handling of a story, that is, the use of certain procedures discernible to the news consumer, protects the newspaperman from the risks of his trade, including critics. [1273, p. 661]

Of course, Tuchman's idea is not unique to news stories, and indeed we might think of all nonfiction as being constituted in a kind of strategic ritual carried out by its author according to social and cultural contexts.[15]

[14]R. Michael Young has written about a kind of "narrative contract" in interactive storytelling that works similarly to the cooperative principal [1370, p. 164].

[15]This notion appears to be very general. Citing Eugene Weinstein [1322], Tuchman writes this also: "Weinstein (1966) speaks of 'credibility enhancing tactics' and suggests (personal

A Fictionality Spectrum

To summarize, scholars have demarcated fiction from nonfiction in three distinctive ways. Caring exclusively about semantic concerns, logicians define nonfiction as having actual referents and true propositional content. Alternatively, formalists seek to identify the endemic stylistic tropes of fiction, such as direct access to character subjectivity and the communication of mental states in a third-person voice; nonfiction, then, could be characterized by the absence of such features. Finally, proponents of a pragmatist approach might argue that nonfiction is produced when nonfiction is intended and nonfiction is received. Again, I refer to these as the *semantic*, *formalist*, and *pragmatist* perspectives, respectively. In the first view, nonfiction depends on the factuality (truth value) of the story, whereas in the latter accounts nonfiction is a *genre*—with stylistic tropes and or a communicative contract between author and audience—in which case factuality may be more or less orthogonal to fictionality.

Note that these schemes do not necessarily depend on binary criteria for nonfiction, which means fictionality may be conceived as a matter of degree. Indeed, several scholars have explicitly stated this [1122, 339, 878]. Marie-Laure Ryan describes such gradation in terms of a two-dimensional plane whose axes pertain to the frequency of referencing an extra-textual world and the implied guarantees about the truth of story content [1068, pp. 179–183]. That paper is in opposition to the postmodernist demotion of nonfiction to fiction, outlined above, which Ryan calls 'panfictionism'. To Ryan, this "doctrine" was successful in bringing about a kind of *paradigm shift* [621] in our conception of nonfiction, but as a totalizing

communication, 1971) tactics connoting 'objectivity,' such as quoting others or presenting alternative possibilities by citing negative evidence or conflicting opinion, may be used to enhance credibility in interpersonal communication" [1273, p. 661]. Such tactics are clearly also germane to academic writing—in fact, I am employing them right now, in this very section (and in this very footnote).

account it does some disservice—here is her lucid articulation:

> The doctrine of panfictionality took us out of the complacency with which we have long approached nonfiction. If structure, tropes, conventions, rhetorical devices and the idiosyncrasies of the medium produce meaning in fiction, so do they in nonfiction. Until the doctrine of panfictionality came along, nonfictional genres of discourse enjoyed an illicit immunity from textual or semiotic forms of investigation. Thanks to the postmodern critique, we are now better aware that texts of nonfiction display an image distinct from their reference world, and that the construction of this image deserves consideration. [...] But the postmodern attack on the dichotomy loses most of its thrust if we give up the simplistic equation of nonfiction with truth and fiction with non-truth. [...] Under the present account, extending the definition of fiction to all discourse would have the negative effect of protecting nonfiction from what I believe are vital forms of inquiry. If fiction has a performative value, it is inappropriate to ask such questions as: "How does the author know?" "How was the data collected and what principles led to its selection?" "Is the interpretation justified by the data?" On the other hand, it is necessary to submit historiography, scientific discourse, literary criticism, or ethnographic writing to this type of scrutiny, just as it is necessary to ask of these genres the question raised by postmodernism: "What kind of paradigm and tacit assumptions led from this data to this interpretation?" [1068, p. 180]

Indeed, it is this nuanced account of fictionality that best characterizes the departure of these scholars from the postmodernist totalizing account of all human endeavor as fiction.[16] In the rest of this section, I will rely on this kind of justification for the existence of a distinct thing that we call 'nonfiction'.

Hayden White's Curation

Lastly, I would like to introduce some ideas from the late Hayden White that will prove foundational to my argument that emergent narrative works like nonfic-

[16] As some have noted, such a totalizing view can have dangerous repercussions: when everything is fiction, no one is beholden to any notion of historical truth, and this may be seen as enabling projects like Holocaust denial [692, 561]. Others, however, argue that postmodernism actually provides a methodology by which such efforts may be adequately confronted: by emphasizing the subjective aspects of historical work, postmodernists are uniquely equipped to expose nefarious aims in ostensibly sincere historical projects [290].

tion.[17] In his seminal *Metahistory* (1973), White characterizes the act of history writing as one of rhetorical curation [1332]. While his arguments are specifically about historiography, I think it would be fair (and useful) to extend them to the broader phenomenon of nonfiction.[18] By emphasizing the constructive nature of historiography, White both destabilizes and augments the predominant and long-held conception of such endeavor, which White summarizes in this way:

> It is sometimes said that the aim of the historian is to explain the past by "finding," "identifying," or "uncovering" the "stories" that lie buried in chronicles; and that the difference between "history" and "fiction" resides in the fact that the historian "finds" his stories, whereas the fiction writer "invents" his. [1332, p. 6]

This earlier view implies that narrative historical accounts exist in an historical record, fully intact, which makes historiography a simple matter of retrieval. To this, the scholar Arthur Danto has articulated a clever refutation in the form of what he call a *narrative sentence*, whereby a description of multiple events is loaded into a sentence about a single event. A canonical example is "The Thirty Years War began in 1618", as Danto explains:

> "The Thirty Years War began in 1618" refers to the beginning and to the end of the war, but it is about the beginning of the war. On the assumption that the war was so-called because of its length, nobody could presumably describe it in 1618—or at any time before 1648—as the "Thirty Years War". Of course someone might predict that the war would last just that long, and put sufficient confidence in his prediction actually to describe the war that way. But he would be making a claim on the future [239, p. 155]

[17]Hayden White passed away in March 2018 here in Santa Cruz, where he was professor emeritus in the History of Consciousness department at UC Santa Cruz. I never met him, and neither did my coadvisors, which means his influence on my thinking in this dissertation is incidental to our living in the same town.

[18]One could argue that nonfiction is necessarily about the past. Even in cases where the present is being documented, by the time the documentation exists, that present has fallen into the past. In any event, this nuance is not important for my argument.

Danto posits this idea in the context of the *ideal chronicler*, an entity that is capable of maintaining a perfect historical record called the *ideal chronicle*:

> I now want to insert an Ideal Chronicler into my picture. He knows whatever happens the moment it happens, even in other minds. And he is to have the gift of instantaneous transcription: everything that happens across the whole forward rim of the Past is set down by him, as it happens, the way it happens. The resultant running account I shall term the Ideal Chronicle [239, p. 152]

As Danto shows, the ideal chronicler cannot construct narrative sentences, because doing so requires foreknowledge of the future, but the entity is omniscient only with regard to the present. Given this, historical records can never contain narrative sentences—but narratologists from Roland Barthes [78] to William Labov [637] show that storytellers tend to cue the causal relations between events. As such, the ramification of Danto's argument is that historical records cannot contain stories. Intriguingly, this suggests a purest distillation of historiography, and all nonfiction, as the matter of crafting narrative sentences.

Hayden White, partly through the influence of Danto [1332, p. 275], does not distill, but rather expands these ideas to describe a more constructive procedure by which subject phenomena are captured in a raw historical record that is then *curated* to produce a narrative historical account with particular rhetorical features.[19] As White explains, this procedure is as much about sensemaking as it is about argumentation:

> [the procedure entails] processes of selection and arrangement of data from the *unprocessed historical record* in the interest of rendering that record more comprehensible to an *audience* of a particular kind. [1332, p. 5]

At the beginning of this procedure, the subject phenomena (constituting an *historical field*) are captured in a *chronicle* that is further curated into a *story*:

[19]To be clear, White does not himself use the term 'curation' (or any other variant), but the procedure that he describes is in line with the sense of the term at work in this dissertation.

> First the elements in the historical field are organized into a chronicle by the arrangement of the events to be dealt with in the temporal order of their occurrence; then the chronicle is organized into a story by the further arrangement of the events into the components of a "spectacle" or process of happening, which is thought to possess a discernible beginning, middle, and end. [1332, p. 5]

In addition to rearranging and filtering the events in a chronicle, constructing a story requires the historian to charge that selected material with rhetorical dynamic. White describes such functionality in terms of *inaugural motifs*, *transitional motifs*, and *terminating motifs*:

> This *transformation of chronicle into story* is effected by the characterization of some events in the chronicle in terms of inaugural motifs, of others in terms of terminating motifs, and of yet others in terms of transitional motifs. An event which is simply reported as having happened at a certain time and place is transformed into an inaugurating event by its characterization as such: "The king went to Westminster on June 3, 1321. There the fateful meeting occurred between the king and the man who was ultimately to challenge him for his throne, though at the time the two men appeared to be destined to become the best of friends...." A transitional motif, on the other hand, signals to the reader to hold his expectations about the significance of the events contained in it in abeyance until some terminating motif has been provided: "While the king was journeying to Westminster, he was informed by his advisers that his enemies awaited him there, and that the prospects of a settlement advantageous to the crown were meager." A terminating motif indicates the apparent end or resolution of a process or situation of tension: "On April 6, 1333, the Battle of Balybourne was fought. The forces of the king were victorious, the rebels routed." [1332, p. 5]

Thus, it is the distinctly narrative quality of the story—marked by the intentional selection of material and the attachment of rhetorical functionality to that selected material—that distinguishes it from the chronicle:

> When a given set of events has been motifically encoded, the reader has been provided with a story; [...] Historical *stories* trace the sequence of events that lead from inaugurations to (provisional) terminations of

> social and cultural processes in a way that *chronicles* are not required to do. Chronicles are, strictly speaking, open-ended. In principle they have no *inaugurations*; they simply "begin" when the chronicler starts recording events. And they have no culminations or resolutions; they can go on indefinitely. Stories, however, have a discernible form (even when that form is an image of a state of chaos) which marks off the events contained in them from the other events that might appear in a comprehensive chronicle of the years covered in their unfoldings. [...] In the chronicle, [an] event is simply "there" as an element of a series; it does not "function" as a story element. The historian arranges the events in the chronicle into a hierarchy of significance by assigning events different functions as story elements in such a way as to disclose the formal coherence of a whole set of events considered as a comprehensible process with a discernible beginning, middle, and end. [1332, pp. 5–6]

White's curation procedure entails three additional steps, which correspond to increasingly higher-order modalities of explanation. In the first, which White calls *emplotment*, a story is cast in the mode ('emplotment') of an archetypical genre (romance, tragedy, comedy or satire). The choice of emplotment affords a particular kind of argumentation, and the afforded argumentation supports a particular ideological position. Finally, by taking this ideological stance, the historian, through the curated history, prescribes a social praxis that the reader may adopt to navigate her present epoch. These final steps in the derivation of an historical account, though interesting, are not particularly relevant to my argument, as I will explain at the end of this section.

To summarize, White describes a procedure by which subject phenomena (an historical field) are captured in a chronicle (an exhaustive historical record) which is then curated into a story that is finally embedded into an historical account with particular rhetorical features.[20] As White states, this curatorial process is precisely what separates historical writing (all nonfiction) from fiction:

[20]In a later book, *The Content of the Form*, White distinguishes between the chronicle and another primitive to the historical account, the *annal* [1331, p. 16]. Relative to the chronicle, the annal has even less narrative structure, and tends to be sparer in its event recording.

> Unlike literary fictions, such as the novel, historical works are made up of events that exist outside the consciousness of the writer. The events reported in a novel can be invented in a way that they cannot be (or are not supposed to be) in a history. This makes it difficult to distinguish between the chronicle of events and the story being told in a literary fiction. In a sense, the "story" being told in a novel such as Mann's *Buddenbrooks* is indistinguishable from the "chronicle" of events reported in the work, even tough we can distinguish between the "chronicle-story" and the "plot" (which is that of an Ironic Tragedy). Unlike the novelist, the historian confronts a veritable chaos of events *already constituted*, out of which he must choose the elements of the story he would tell. [1332, p. 6]

By now the connection of White's ideas to my claim in this section, and to the larger framework of this dissertation, should be clear: curationist emergent narrative is constituted in the same curatorial process that White attributes to historiography, but instead of taking real-world phenomena as its subject material, it takes simulated phenomena.

Before moving on, I should mention that others have made the connection between Hayden White and procedural narrative [42, 126, 1096, 234]. Here, of particular relevance is a fabulous paper (that I will reference several more times) in which Stephanie Boluk and Patrick LeMieux argue that some of *Dwarf Fortress*'s [17] generated material resembles a Whitean chronicle:

> *Dwarf Fortress*'s linguistic forms of historical inscription in Legends mode bear a striking formal and thematic resemblance to early forms of writing, such as the medieval annal and chronicle. [126, p. 144]

I will now proceed to argue that (curationist) emergent narrative works like nonfiction by discussing an example of the form in terms of the definitions of nonfiction that I have provided so far.

Case Study: *Oilfurnace*

Let us finally return to the thesis of this section: emergent narrative works like nonfiction.[21] To ground this discussion, we will consider the case of *Oilfurnace* (2010), a narrative comic by Tim Denee that recounts the emergent events of a *Dwarf Fortress* [17] gameplay session. The comic was commissioned by the Australian magazine *PC PowerPlay* and appeared in its June 2010 issue [255]; an excerpt is shown in Figure 3.1.[22] *Oilfurnace* recounts a *Dwarf Fortress* gameplay session that Denee carried out over the course of two or three months [756], and as such it is clearly a work of emergent narrative—and more particularly of curationist emergent narrative, as I will emphasize in Chapter 5.

Oilfurnace begins by explaining that the titular Oilfurnace is a dwarven colony established on a place called Dread Island, a barren isle in the region of The Fatal Forests, by the decree of a dwarven king in the (dwarven) year 1050. The colonists are The Fatal Forests Trading Company, a troupe of seven individuals of varying specializations—four are miners, the others are multiskilled—who are led by a character named Ast Akrulikal. Prior to Oilfurnace, four earlier colonies had been attempted at the site, but each ended in disaster: zombie gorillas, thirst and starvation, zombie elephants, and flood.[23] From this background, the comic proceeds to tell the story of the colony in terms of four distinct periods in its history (each taking up its own page): The Foundation, The Rise, The Golden Age, and The Fall.

[21] My argument in this section was bolstered by conversations with Max Kreminski and Jason Grinblat, who each take a similar stance on these issues.

[22] The entire comic is freely available online at `https://www.timdenee.com/oilfurnace` [256]. Prior to *Oilfurnace*, Denee produced a similar comic called *Bronzemurdered* [254], and he is also the illustrator of *Getting Started With Dwarf Fortress* [1281]. Many more stories recounting *Dwarf Fortress* gameplay are available on the website *Dwarf Fortress Stories* [1287], which is curated by Taran and Trevor Van Hemert.

[23] This narration is an allusion to the emergent narrative of earlier gameplay sessions carried out by Denee in the same simulated storyworld.

Figure 3.1: An excerpt from *Oilfurnace* (2010), a narrative comic by Tim Denee that recounts the emergent events of a *Dwarf Fortress* gameplay session. In this final segment of the comic, the previously thriving Oilfurnace colony opens a gateway to the underworld and faces the consequences thereof. This curated story, I contend, works like nonfiction, particularly in the sense of Hayden White's historiography.

In Denee's narration of the fist period, we learn of the foundation of the colony through an ordered series of discrete events that are recounted in comic panels that each couple an illustration and a sentence or two of descriptive natural language. These pertain primarily to the successful construction of an elaborate shaft allowing access to the rock layer below an aquifer that had brought the demise of two of the earlier colonies (in one it could not be traversed, in the other it flooded). The narration of this period culminates in a miner discovering platinum in the rock layer below the aquifer.

At the beginning of the next period, the dwarves dig into the dry stone beneath the aquifer and carve out a subterranean fortress. From here, industry proceeds—trees are harvested, ore is mined, coal is burnt, metal is smelted—and new migrants arrive at the island fortress by boat. An invasion of undead slugmen and snailmen challenges the fortress, but a militia comprising Ast Akrulikal and a dwarf dubbed 'Batdwarf' (he wears a robe made of bat leather) swiftly disposes of them. More migrants come and so do more invaders, but the militia grows in turn and these challenges are also overcome. A period of starvation ensues, but this is solved by the development of a series of cave farms in which mushrooms are cultivated. At this point, precious metals abound in the fortress and a first mayor is elected. As a final hardship in this period, troglodytes begin to emerge out of the fortress mine shafts, but a series of traps are constructed accordingly. Finally, a shaft is dug deep, into a magma layer in "the bowels of the earth", and the dwarves use this as a garbage disposal. The period culminates, as before, in the discovery of a precious mineral—this time, miners encounter adamantine at the bottom of the magma trash chute.

From here, Denee proceeds to narrate The Golden Age of Oilfurnace. On this page, the explanation is more visual than narrational, illustrating the layout of the fortress and concerns such as the processes that characterize its steel and adamantine industries (which emerge from the game's *crafting system* [431]). Additionally, we learn that the six surviving members of the previous period's militia have each attained the status of 'legendary killer'—this cohort is now dubbed the 'zombie slayers', as they deal swiftly with a series of zombie hordes. A new generation of dwarves is growing up in the fortress, the population is booming, and an arena for entertainment and training has even been constructed. A foreboding panel at the bottom of the page, however, notes the presence of five *forgotten beasts* [1281, p. 172] in the magma layer below the fortress.

Finally, Denee narrates the closing period in the history of the colony: The Fall of Oilfurnace. Here, he returns to a predominantly narrational style, with ordered event panels; this page is shown in Figure 3.1.[24] The period begins with the fortress mayor demanding, hubristically, that coins be minted out of pure adamantine, which requires miners to dig near the deepest parts of the magma sea below the fortress. Unfortunately, this opens a gate to the *underworld* [283]. From here, the narration tightly reconstructs a rapid series of events that begins with a group of demons pouring out of the gateway to meet the zombie slayers. The demons, Denee explains, are too powerful and too numerous: members of the militia perish, and Batdwarf is forced to sacrifice himself to allow Ast Akrulikal the time and space to retreat from the depths. Miners are ordered to breach the aquifer (to drown the demons), but they panic or die before this can be done. An alarm bell sounds to call for evacuation, and Ast (with no weapon and a broken arm) and six other survivors manage to escape and close the fortress gate behind them. With the dwarves now outside and helpless, a band of zombie slugmen emerge from the dead woods of Dread Island to descend upon them.

[24] Note that the temporal sequence is made explicit by numbering each panel; this device was also used on the second page.

Only Ast survives the onslaught—"he punches a slugman through the head"—and he retreats to a secure building in the woods. In this secret keep, Ast is shown reading a letter from the dwarven king that reads *Remember: losing is fun* (panels 17–18)—this is the *Dwarf Fortress* community's favorite slogan [1281, p. 2]—and finally he pulls a "self-destruct lever", thereby locking himself safely (but permanently) inside. In a final panel, Ast is shown rotted to bones in the keep, and Denee concludes the tale:

> The entrance to the fortress is blocked by pulling the self-destruct lever. With nothing waiting for him outside the lever-keep, Ast waits for death inside, away from the harsh glare of the overbright. If a dwarf cannot die underground, at least let him die with a roof over his head. A noble end to an ignoble fortress; it is all any dwarf can hope for.

***Oilfurnace* as Nonfiction**

I contend that *Oilfurnace* is a work of nonfiction (or at least works like one). To explore this claim, let us treat the descriptions of nonfiction given above as classification schemes: if they can be applied to *Oilfurnace*, then its status as nonfiction is supported, and from here it is reasonable to extrapolate this characterization to other works of emergent narrative.

First, it is prudent to consider the objections to the very notion of nonfiction that I have outlined above. If Gorgias, taken seriously, is correct, then nothing exists: there is no *Oilfurnace*, no dissertation, no me, no you. If one allows for solipsism, then still I cannot be sure that *Oilfurnace* exists and neither can you. Moving beyond these troubles, we encounter the postmodernist assassination of objectivity. In the case of the artifact at hand, this perspective might lead us to maintain that, in constructing a comic that recounts the emergent events of his *Dwarf Fortress* gameplay session, Tim Denee has produced a work of fiction, because it impossible to produce an objective account of anything.[25] This is a reasonable view that I accept, but like the various other scholars who were discussed above, I think it is also reasonable to say that objectivity is not a necessary feature of nonfiction. That is, there is an identifiable genre called nonfiction, and it persists in spite of the impossibility of human objectivity. This, again, is the sense of 'nonfiction' that I appeal to in making this argument, and so the postmodern critique is not a showstopper here.[26]

Before moving on, let us consider the postmodernist critique as it would apply to emergent narrative as it more commonly appears: an unfiltered stream of phenomena that transpires over the course of an interaction with a simulation. In Chapter 4, I will build on Espen Aarseth's critique of this approach to emergent narrative (my view: narrative only obtains when the raw stream is curated), but nonetheless it is currently the predominant conception of the form. As such, it is interesting to ponder: are these streams of raw emergent narrative objective? It turns out, I think, that this consideration does not really compute, since it is like asking whether the raw stream of reality is objective—we can only know it through a subjective experience of it, and the artifact of our encounter (whether that be a mental understanding or a work of media) is thus always subjective. In the case of a raw stream of simulated material, we might say that the narrative obtains in the mind of the interactor as the stream transpires, but then the result is clearly subjective. Really, this consideration does not make much sense because it requires us to ignore that such streams lack the narrative structure that characterizes the fiction and 'nonfiction' artifacts that are the subject of the postmodernist critique

[25] Barthes, in a similar vein, might specifically argue that the sign system of the comic has as its signifieds not the actual characters and material of the Oilfurnace colony and its larger simulated world, but rather simulacra of those entities. That is, the denotational procedure constituted in the composition of any human communication, including a comic, makes objectivity impossible.

[26] To be clear, this is what I mean when I say that emergent narrative 'works like' nonfiction.

(which means we cannot proceed in good faith). This conceptual trouble is an expression of what I view as the fundamental issue with emergent narrative: raw event streams are not narrative, but for some reason they have been viewed as such in the particular case of simulation.

Semantic Perspective: *Oilfurnance*'s World

Next, let us consider the *semantic perspective* on fictionality. According to the view articulated by David Lewis, nonfiction obtains when (or to the degree that) a narrative asserts true propositions about real referents. Who are the referents of *Oilfurnace*'s narration? They include characters such as Ast Akrulikal, Batdwarf, and other named dwarves; unnamed characters including the various zombies and forgotten beasts; locations such as The Fatal Forests, Dread Island, the Oilfurnace colony and fortress, and components of the fortress; the named earlier attempts at colonizing Dread Island; and many more. Logical propositions about these entities, then, are asserted through the various storytelling mechanisms employed in the comic: visual images, natural language prose, panel arrangements, and any other technique one might encounter, for instance, in the poetics of sequential art proffered by Will Eisner [304] or Scott McCloud [796] or others [545]. So, are these real referents and true propositions?

This line of thinking ultimately converges on a discussion of the ontology of simulation, and in particular whether simulated entities and events are realer than their counterparts that obtain in conventional fiction. One might contend that a simulated entity is not real because it is virtual, not physical. To this, another might counter that the simulated entity *does* indeed have a physical constitution, one that obtains in the material of the computing machine. This is probably not a great argument, however, because that physical constitution will vary across runtime instances, even if the virtual entity remains identical (by some policy for determining isomoprhism).

Alternatively, one could say a simulated entity is not real because it is a simulacrum. But if Ast Akrulikal is a simulacrum, what is the original entity that it represents? Due to the use of world generation in *Dwarf Fortress*—i.e., due to its *roguelikeness* [379]—Ast Akrulikal only exists in the storyworld that is associated with the particular world seed and software version that Tim Denee used in his gameplay session. Thus, Ast Akrulikal is no simulacrum: he is not a human being, or even a being of any sort, but he is something of a particular kind—a *Dwarf Fortress* dwarf—and he exists in his world at the full fidelity of existing as that kind.[27] That is, he could not be more perfectly modeled, just like you could not be more perfectly modeled—you are the perfect representation of yourself, because you are yourself, and the same is true of this character.

One might argue that this does not hold for characters in conventional fiction: the Sherlock Holmes that one encounters in print, for example, is a lossy representation of the pure Sherlock Holmes that existed in the mind of Sir Arthur Conan Doyle, making the former a simulacrum of the latter.[28] Intriguingly, this situation only obtains in roguelike procedural narrative—an approach often taken in works

[27]More specifically, this kind is a *Dwarf Fortress* dwarf in the particular software version that Denee executed. When Tarn Adams changes game code that affects the representation or behavior of dwarves (or of any part of the gameworld that affects dwarves in any way), then the ontology of the *Dwarf Fortress* dwarf changes. This means that Ast Akrulikal and a dwarf of another version of the game may be of different kinds. As Ian Horswill noted in his feedback on an earlier draft of this thesis, the treatment of simulated entities as being real raises peculiar ethical questions: "If DF dwarves are real, what are the ethics of killing them?" (personal communication, July 28, 2018). I will not dive into this issue here, but for discussion of it with regard to *The Sims* [792], for instance, see this paper by Juyun Kim and Stephen Petrina [579].

[28]Interestingly, since Doyle is dead, one could state that the original is gone, and all that is left is the copy. This calls to mind Jean Baudrillard's writing, in *Simulacra and Simulation* [88], about copies with no (extant) originals. As Noah Wardrip-Fruin noted in his feedback on an earlier draft of this thesis, this argument is made tenuous by the shift in literary studies away from the author (e.g., [77]). Indeed, the argument will fall apart by the end of this paragraph, but I think the idea is still worth mentioning.

of emergent narrative—since authored characters still appear in other kinds of computational media. For instance, my claim about Sherlock Holmes could be applied to the *Façade* character Trip [778], though now the original(s) would exist in two minds (belonging respectively to co-creators Michael Mateas and Andrew Stern). In this way, the argument begins to fall apart. Moreover, even in the case of Ast Akrulikal, another player could access the same storyworld by using the same seed and software version (or Denee himself could, for a second time), and assuming divergent gameplay inputs, the world would proceed differently and Ast Akrulikal himself would likely change as well. Generally, I think it is probably futile to make ontological claims about the realness of simulated entities (and even if realness could be established, issues with objectivity could prevent direct access to the true forms).[29]

Instead of considering the ontology of simulated entities, one might explore an intuition: simulated events seem to *happen* in a way that authored fictional events do not. That is, the stuff of emergent narrative is material that is not authored, but instead emerges out of the complex interaction of authored processes. I am not sure how to construct a formal semantic account of this intuition, since the representation of true propositions referring to nonexistent entities would require a peculiar logic. Instead, I will contend that this intuition, and the feeling

[29] Again, Barthes's reference illusion comes to mind: perhaps Ast Akrulikal is merely a kind of signified with no referent. I will note also that others have made ontological claims about expressive simulations. Cameron Kunzelman postulates, evocatively, that the system of a videogame constitutes a real, living body that humans live both with and within [624]. Several thinkers have suggested that the gameworlds of massively multiplayer games like *Eve Online* are real worlds [167], though this idea is bolstered by fact that many human interactors pilot characters in its multiplayer networked storyworld simulation. In a distinct but related vein, Stephanie Boluk and Patrick LeMieux describe simulated storyworlds as monuments to the real mechanisms that produce them: "Despite its minimal textual interface, the process of generating this history weighs heavily on the central processing units of most computers. The millions of events logged during world generation are granular enough not only to correspond to the history of the gamespace represented on-screen but also to ultimately historicize the processor cycles of the computer itself" [126, p. 126].

that procedurally generated entities are somehow realer than authored fictional characters, supports a loose idea that emergent narrative works like nonfiction. Whether this is true or not, the notion, when held by an author or reader, supports a particular set of aesthetics—indeed, ontology may be orthogonal to aesthetics, since the aesthetics of a work for an individual depends on her conception of that work.[30] In Section 3.2, I discuss the aesthetics supported by this intuition.

While I find this intuition alone to be powerful, I will now provide a concise logical account of *Oilfurnace* as nonfiction that is rooted in John Heintz's identified features of fictional worlds. As outlined above, Heintz describes fictional worlds as being logically incomplete and logically inconsistent, and thereby he implies a definition of nonfiction as narrative that recounts complete and consistent worlds.

Due to the nature of computer simulation, the simulated storyworlds of emergent narrative are in fact complete and consistent. As an example, let us consider the case of the *Dwarf Fortress* world that *Oilfurnace* recounts—specifically, let us discuss logical *decidability* (whether and how one can identify truths and non-truths) with regard to this world. As the world obtains through the execution of the game's software, a set of facts about the world will be asserted as an unavoidable byproduct of that execution. These facts will pertain to concerns such as entity attributes, the order of simulated events, details of the physical world model, and much more. Some of these facts will be stored in computer memory, which means propositions about the storyworld may be evaluated by querying against this data. As such, the notion of a truth (pertaining to the simulated storyworld) may be operationalized as a fact that can be queried during execution of the game's software (of course, with the seed and software version that

[30]In my project *Sheldon County*, discussed in Chapter 12, I specifically frame the media artifact as a work of nonfiction (both in the work itself and in supporting materials). Whether that is true or not, it is how I conceive of the work, and by framing it accordingly, listeners will be led to also view it in this way.

indexes that world being used).[31] Note that some truths about the world will not be recorded as persistent facts that are stored in data, because it is not feasible to record all data about the gameworld—we might call these *ephemeral facts*. Nonetheless, even an ephemeral fact qualifies as constituting a truth under this operationalization scheme, because the fact could have been queried in the brief moment that the pertinent data existed in memory. As such, all truths about the storyworld will be rendered in asserted facts as the simulation proceeds. That is, with regard to a simulated timestep t, all truths about the world that hold through t will have been asserted in the execution to that point, and no truth that holds at t will not have been asserted.

Still, there is an outstanding issue that pertains to the notion of decidability in the *Oilfurnace* storyworld: we may wish to evaluate certain propositions for which corresponding queries cannot be formed. This would occur when an attempt is made to evaluate a proposition that depends on something that is not modeled in the storyworld. For example, to adapt a predicate that Daniel Dennett has applied to Sherlock Holmes [257], we might wish to ask whether Ast Akrulikal has a mole on his left shoulder blade. While *Dwarf Fortress* is famous for its detailed modeling (e.g., of organ tissue [282]), I do not believe such a query could be formed, since the simulation does not represent moles and thus the software would not "understand" the query. More precisely, `false` would not be returned—as it would be if a query asked whether Ast's name is 'Batdwarf', for instance—but rather some kind of error would occur, since the query would not be well-formed with regard to the game's data representation. One might call the propositions corresponding to such

[31] As Ian Horswill noted in his feedback on an earlier draft of this thesis, in the *Versu* project Richard Evans and Emily Short very deliberately model the storyworld as a set of facts (asserted in a modal logic). As they argue in a paper on the project, this design method produces a number of authorial and architectural affordances, namely the ability to "find out what is true" [326, p. 118].

queries *undecided*, which would mean that the *law of bivalence* (all propositions are either true or false) does not hold for *Dwarf Fortress* storyworlds, which would mean that they are incomplete.[32]

Alternatively, however, it is tempting to say that in such cases Ast does *not* have a mole, because such a mole is not modeled—that is, if a query cannot be formed, the negation of the corresponding proposition is implied. Is this fair, though? One could just as well say that anything about Sherlock Holmes that is not explicitly stated in the text of Doyle's stories is false. To me, however, it feels more intuitive to allow this in the case of a simulated world whose ontology is explicitly represented as structured data. Additionally, while Doyle could have written a new installment about the mystery of the mole on Holmes's left shoulder blade, thereby rendering the proposition's truth value decided, this cannot be done in the case of *Dwarf Fortress*: when the code that produces a given storyworld is changed, the world becomes inaccessible.[33] Thus, I contend that the world of *Oilfurnace* is complete, or at least that it *feels* complete: even if a detailed consideration reveals that the law of bivalence does not hold in computer simulation, it *feels* like it does. This feeling, which I will continue to validate throughout the rest of the section, is the bedrock that supports the notion that emergent narrative works like nonfiction. Even if one takes the ontological position that simulated storyworlds are not consistent, they still feel consistent, and this supports a particular set of aesthetics—these are the aesthetics of emergent

[32]To be fair, it is not clear that this law holds for the real world either [1285, 840], as Ian Horswill noted in his thesis feedback.

[33]I admit that it would be possible for Tarn Adams to alter the code such that the storyworld remains accessible and intact, except for the advent of moles, through a clever engineering of the world-seeding functionality. Indeed, Adams spends considerable effort on ensuring backward compatibility in the game, which means storyworlds from old versions (stored as save files) can be used in newer versions [284]. That being said, what I am attempting to elicit here is a particular intuition: simulated storyworlds feel like closed systems, and fictional worlds less so.

narrative, which I outline in Section 3.2.[34]

Due to the nature of simulation and of computation more broadly, the simulated storyworld of *Oilfurnace* is also *consistent*. Recall that Heintz characterizes fictional worlds as being inconsistent when incompatible propositions are asserted—for instance, *Anna Karenina* implies both that the events of its story begin on a Monday and on a Tuesday. At a certain level of detail, this kind of inconsistency is impossible in computing, since it may be seen as requiring incompatible data to be stored in the same memory locations.[35] This points to a fundamental difference between a fictional world and a simulated world, which is that the latter emerges from a system of laws (the mechanics of the simulation), whereas the former obtains through a kind of human artifice. Of course, in simulation a human may still author the system of laws, but upon being authored those laws constitute a closed system that is consistent. It may not be consistent with the real world or whatever it may be modeling, but it is nonetheless an internally consistent system that is complete with regard to itself.[36] Thus, when it comes to consistency, human invention is mutable, while simulation tends not to be.[37]

[34]As I have already explained above, my argument here is fundamentally aesthetical in nature, rather than ontological or even philosophical: I am interested, foremost, in how emergent narrative feels how it feels, and why humans like that feeling.

[35]This kind of *superposition* is actually possible in *quantum computing* [433], but the ontological implications of that paradigm extend far beyond our concerns here.

[36]I should mention a counterexample here. In his independent and collaborative doctoral work [1014, 1012, 1011, 1013], Justus Robertson has explored how incompatible assertions about a storyworld can be maintained as equally valid, using a metaphor of *superposition*. Specifically, a superposition (set of competing assertions) may collapse (onto a sole assertion) in accordance with a human observer's experience of the world. For example, an observer might encounter two doors, one on the left and the other on the right, whose respective connecting rooms have not been decided by the system. As the observer enters the left door, a connecting room is decided for it, and from now on the door on the left will always connect to that room, while the door on the right will not. Alternatively, had the observer entered the right door, *it* would have connected to that room and the left would not have. The ontological and aesthetic ramifications of Robertson's approach are fascinating and worthy of extensive discussion, though unfortunately that is also beyond the scope of this study.

[37]In his notes on an earlier draft of this thesis, Ian Horswill identified an important exception here: "there are lots of cases of games in which there is a detailed simulation, but there's something that appears in the game that's referred to, either in the dialog or in the visuals, that

It is worth addressing the potential counterclaim that these properties of completeness and consistency, to the degree that they are based in ontology, in fact apply to all works of computational narrative (i.e., even ones outside the paradigm of emergent narrative). Indeed, in each case a storyworld will obtain through the execution of a computer program. However, the primary difference, I contend, originates in the hallmark of emergent narrative: it is driven bottom-up by simulation, not top-down by narrative. In emergent narrative, the rules of the simulation are the laws that govern the storyworld itself, while in conventional computational narrative the rules of the simulation govern the narrative, the instantiation of which may *suggest* a larger storyworld in the same way that the prose of print fiction suggests a larger storyworld. For instance, consider the case of *Façade* [775]. There is a simulation at work—it models the physical environment of the apartment, the hallway outside it, and apparently the balcony outside [898]—but it does not model the entirety of the storyworld that the unfolding narrative implies. For instance, past events (such as a vacation in Italy) are implied, but these events are not actually simulated. The history of the world of *Oilfurnace*, on the other hand, was actually simulated prior to the beginning of gameplay—this is *Dwarf Fortress*'s famous *world generation* procedure [442, 751]. While all simulations have gaps (see Section 4.1.5) and imply extraneous material, this occurs less frequently as the level of simulation increases, and the hallmark of emergent narrative is intensive world simulation. As such, the storyworlds of emergent narrative feel more complete and more consistent, and so emergent narrative works more like nonfiction.

isn't modeled in the world" (personal communication, July 28, 2018). This is a great point, and one that further erodes the basis for any argument that simulated storyworlds are real in an ontological sense. As I will explain more at various points below, my ultimate aim here is not to make ontological claims, but rather to argue in support of an artistic position: regardless of their true ontology, simulated storyworlds *feel* real, and this produces a distinct pleasure.

Formalist Perspective: *Oilfurnace*'s Tropes

As I outlined above, the *formalist perspective* on fictionality ignores the logical status of narrative content to instead focus on the stylistic tropes that characterize works of fiction. Identified tropes include a means of entering the subjective worlds of characters and surfacing the internal concerns found therein, often in a third-person voice. Intriguingly, the simulation of characters in a storyworld may easily support this phenomena: a system containing such a simulation will be omniscient with regard to it, in the sense that any truth about the simulation may be accessed through querying (as discussed above). Indeed, in my own work, I am deeply concerned with modeling of the internal lives of synthetic characters, and ultimately my goal is to surface interesting internal phenomena that emerges through such modeling. Does this make emergent narrative more like fiction, then? Not necessarily, I contend. First of all, nonfiction, as a genre of writing, is rarely divorced from the internal worlds of its characters (real individuals). Even beyond *new journalism* [1350],[38] classical 'objective' nonfiction writes about the internal worlds of individuals, sometimes as a byproduct of indirect access (e.g., through someone's diary writings), but often through a rational postulation. In any event, the only impediment to this direct accessing for the nonfiction writer is her lack of omniscience. But if an omniscient being wrote nonfiction that delved with perfect accuracy into the mind of a real individual, we would not say that such access turns that account into fiction.

As such, though the tropes of fiction identified by this formalist view may apply to works of emergent narrative, this does not mean they do not work like nonfiction. Moreover, through its characteristic omniscient access to a storyworld, emergent narrative enables a unique brand of nonfiction that is in fact only made

[38]In the midst of writing this paragraph, I learned that Tom Wolfe, the figurehead of the new journalism movement, died today. Rest in peace.

possible through something like computer simulation. Indeed, the very possibility of omniscience in emergent narrative may be viewed as an argument for why it works like nonfiction. Again, however, this is a feature that is also present in other works of computational narrative, though again to a lesser degree corresponding to the reduction in explicit modeling of a storyworld.

Pragmatist Perspective: ***Oilfurnace*'s Contract**

According to the *pragmatist perspective*, fictionality depends on the *communicative intent* of the author and a kind of social contract that is consummated between author and audience. That is, nonfiction obtains when an author intends to produce nonfiction and the audience agrees that the artifact composed thereby is a work of nonfiction. Thus, nonfiction is made possible by a *shared pretense* [1261]. Here, we might consider a pivotal factor in the interpretation of *Oilfurnace*: whether the audience is aware of *Dwarf Fortress* and the comic's relation to it. To the uninitiated, the comic *constructs* a narrative set arbitrarily in a fantasy world, while to the initiated it *recounts* a narrative that emerged out of the game's famous simulation. Indeed, while I have not encountered the comic except as someone who was already aware of *Dwarf Fortress*, I suggest that the comic, as a standalone artifact divorced from any understanding of the game, is a strange artifact. Its narrational style is peculiar, and the quality of the narrative is perhaps even dubious. If Tim Denee were to write a story from whole cloth, he might produce something more extravagant—the quality of this story, however, is dependent on an understanding that the events that it recounts *actually happened*, in the sense that they emerged out of the complex interaction of processes in a computer simulation. As I will express more thoroughly in the later sections of this chapter, a story that recounts actual events will tend to be inherently more interesting than a hypothetical fictional counterpoint with the same content. This is the basis for 'based on a true story'—it is what produces the *pleasure of nonfiction*.

In the case of emergent narrative, this means that the stories generated by this approach receive a boost in interestingness *when the audience is made aware of the simulational origins of the content*. To the degree that the simulation is inhibited through interventionist techniques, such as drama management or narrative planning, this interestingness boost dissipates.[39] One might even say that procedural narrative works like nonfiction to a degree that is commensurate to this interestingness boost. It is critical, however, to realize that the boost is not actuated merely by the simulational origins of a generated story, but by a *contract* between author and audience that transacts interestingness in exchange for guarantees about the 'actuality' of the recounted events—this might be called the *contract of emergent narrative*.

The primary takeaway here is that works of emergent narrative must do some work to emphasize the actuality of the recounted simulated events so that a contract, transacting such actuality in exchange for a boost in perceived interestingness, may be consummated between the author and the audience. This emphasis may be carried out in the content of the work itself, or in the *paratext* [387], meaning any external materials that frame interpretation of a work. In Chapters 10 and 12, I discuss how this contract is consummated in *Bad News* (primarily through its preliminaries and epilogue) and in *Sheldon County* (primarily through its paratext and the opening segment of each generated pilot episode).

[39] I will expand on this argument in Chapter 5.

Whitean Perspective: ***Oilfurnace*'s Curation**

Finally, let us view *Oilfurnace* through the lens of Hayden White's model of historiography as a procedure of curation. By this procedure, which entails three distinct phases, subject phenomena of interest to historians are eventually recounted in historical accounts with particular rhetorical features.

First, the subject phenomena are captured, as they are transpiring, in a *chronicle* that records them by some method of inscription. While the chronicle itself is the result of a kind of curation, since it will never perfectly capture the subject phenomena (inscription is lossy), it is, for the historian's purposes, the raw historical record. That is, because it is the only documentation of the subject phenomena, it contains the only material that the historian may use to construct her account of that phenomena (though she of course may augment this, or adulterate it, with extraneous material). The chronicle is open-ended, with no narrative beginning (but rather the unceremonious onset of a process of recording that produced the record) and no narrative end (but instead an unceremonious termination of that recording process). From this chronicle, the historian crafts a *story.* This process entails the selection of a subset of the chronicled events, which may then be used to construct a discernible narrative structure in which some of selected events actuate *motifs*: *inaugural motifs* cue meaningful causal sequences that will culminate eventually with *terminating motifs*, and along the way *transitional motifs* signal abeyance in the causal sequence. Finally, the story is embedded in a particular *emplotment*—e.g., tragedy or comedy—which unlocks a set of rhetorical effects that support a targeted ideological stance.[40]

Let us now consider analogues between the components of emergent narrative

[40] As I have noted already, while White delves into considerably more detail as to the latter stages of the historiographic process [1332], it is the initial steps that are of particular concern to our purposes here.

and White's notions of an historical field, chronicle, story, and emplotment. Again, we will consider emergent narrative through the concrete example of *Oilfurnace*.

First, what I have called the 'subject phenomena' of a story, White terms the *historical field*. In *Oilfurnace*, this corresponds to the unfiltered stream of simulated phenomena that transpired during Denee's gameplay session. For the most part, this phenomena is ephemeral, since it would be infeasible (from the standpoint of computer memory) for the system to record the whole plenitude.

As for White's 'chronicle', we might identify two analogues in *Oilfurnace*. First, whatever record keeping the system *does* manage to do (in terms of recording simulated phenomena) will result in a kind of chronicle (in the form of stored data). Second, Denee himself likely maintained a kind of chronicle in the form of notes or other inscriptions (whether physical or digital) recorded during the course of gameplay.[41] Note that the existence of multiple such records poses no trouble to this analogy, since for many historical concerns the notion of a chronicle takes the form of a composite record that combines information from multiple sources. Thus, we may say that in the case of *Oilfurnace*, these records combine together to form a composite chronicle.

Next, we may identify as an analogue to White's 'story' the subset of material from this chronicle that Denee selected for expression in the comic. Here, we even find White's 'motifs': an early panel reading "Strike the earth!" inaugurates the

[41] We might also posit that Denee's own memory of the gameplay session is a kind of chronicle. I am not sure whether this idea would be in accord with White's thinking, but this may be due to the fact that he describes the particular case of historiography, and typically the historian is personally (and temporally) removed from her subject phenomena. As I articulate below, while I am interested in human-curated works like *Oilfurnace*, I am especially interested in the prospect of automated systems that procedurally curate simulated histories. This is what I am exploring now with my project *Sheldon County*, and as I explain in Chapter 12, I have found that such automatic curation is greatly assisted by the simulation maintaining extensive records of its generated phenomena. In this configuration, where one computer program is curating the material inscribed in a record kept by another computer program, we find an uncanny analogy to White's notion of the historian who constructs a story by curating a chronicle.

start of an extensive dig that terminates finally in the fatal demon invasion that was caused by the dwarves (hubristically) digging too deep. Along the way, a series of transitional motifs reference the discovery (through increased digging) of increasingly appealing minerals.

Lastly, White's 'emplotment' mechanism appears in *Oilfurnace* through its evocation of the tragedy, which is signaled frequently: a sense of doom is instilled immediately through the initial reference to four failed earlier attempts at colonizing Dread Island; new challenges are encountered and overcome, but victory is always tenuous; ultimately a final challenge (the demon invasion) cannot be overcome, and everyone perishes; the king's emergency letter, opened by Ast as a last resort, reminds the reader that "losing is fun".[42] Because there are different stakes to the rhetoric of historiography and to that of emergent narrative, White's articulated concerns at the level of emplotment and beyond become less relevant to our concerns here.[43]

Clearly, analogies abound between White's model of historiography and the process by which *Oilfurnace* would have been created. While White's greater aim is to show that historiography is primarily a rhetorical practice, the process

[42]Indeed, this slogan and the tragic fatalism that it celebrates suggests that the emplotment of *Dwarf Fortress* gameplay itself is the tragedy.

[43]This is not to say that it would be impossible to curate emergent narrative to particular rhetorical effect. Indeed, my initial dissertation topic [1038] was a kind of story generation by which characters situated in a storyworld would curate their accumulated knowledge of that world (a kind of chronicle) to tell stories to one another to particular rhetorical effects. That is, it would be *pragmatic* story generation, whereby characters storytell as a way to change the world. Moreover, it is also certainly possible to do curationist emergent narrative in a way that would target a particular ideological stance (with regard to the real world, not just a simulated one). This is essentially one of the aims of my colleague Melanie Dickinson, who is exploring of use of social simulation for social justice [263]. Though not a work of emergent narrative, *Terminal Time* [768, 780, 268] is a remarkable example of procedural nonfiction that explicitly targets particular rhetorical effects. In the project, an AI system works to generate a documentary film that exaggerates an audience's collective stance on an issue to the level of a lampoon. Michael Mateas, who is one of its cocreators (along with Steffi Domike and Paul Vanouse) and also my coadvisor, recently told me that several individuals who experienced the piece remarked that it was essentially operationalizing the ideas of Hayden White.

of curation that he describes signals a critical distinction between fiction and nonfiction. To emphasize this point, let us again consider White's articulation on the matter (which was already quoted earlier):

> Unlike literary fictions, such as the novel, historical works are made up of events that exist outside the consciousness of the writer. The events reported in a novel can be invented in a way that they cannot be (or are not supposed to be) in a history. This makes it difficult to distinguish between the chronicle of events and the story being told in a literary fiction. In a sense, the "story" being told in a novel such as Mann's *Buddenbrooks* is indistinguishable from the "chronicle" of events reported in the work, even tough we can distinguish between the "chronicle-story" and the "plot" (which is that of an Ironic Tragedy). Unlike the novelist, the historian confronts a veritable chaos of events *already constituted*, out of which he must choose the elements of the story he would tell. [1332, p. 6]

As I have noted above, I think it is fair (and valuable) to consider White's model as a description not just of historiography, but of nonfiction more generally. In taking this broader view, we find an even stronger account of how emergent narrative works like nonfiction: in fiction and other forms of computational narrative, the subject matter of a story is instantiated *through the process of narration*, whereas in emergent narrative these events are *already constituted at the time of narration*.[44] Note that this is especially the case in curationist emergent narrative, where effort is dedicated to the construction, through curation, of an actual narrative artifact. Indeed, I contend that this curationist approach does better than any other method for emergent narrative to unlock the pleasures of nonfiction and the aesthetics that result thereby. Let us conclude this section with a quote from Tim Denee, in which he himself describes the creative process

[44] In his comments on an earlier draft of this thesis, Ian Horswill wrote: "This is a great model for Oilfurnace, but DF itself doesn't have a curator. So would you then say that a person just 'playing' DF, and not curating a story about it for a third person, was not experiencing emergent narrative?" I address this matter extensively below, in multiple sections, so if you are wondering the same thing, hold tight or else jump to Section 5.3.2.

that produced *Oilfurnace* as one that resembles the curation procedure described by Hayden White:

> I think Dwarf Fortress generates such complexity and depth of information, that the task of the storyteller is to simply sift through and find the narrative [**?**, n.p]mastrapa2010kiwi

3.1.2 Analogy to Stories of Lived Experience

In the fall of 2002, I was fifteen years old and deeply immersed in *Grand Theft Auto: Vice City* [883]. The game was enjoyable to me not for its notorious content and violent gameplay, but for its simulation of a vibrant bustling city. Earlier open-world simulation games had typically been set in fantasy or science-fiction environments, but this game was set in a version of the real world, which appealed more to my tastes.[45] Most importantly, the game's emergent possibility space seemed marvelously vast. I felt that I could navigate this space in a constructive manner—like I was crafting a story through the way that I played.[46]

Eventually, I became disenchanted by the hollow modeling of non-player characters (they do not afford meaningful interactions beyond different types of collision) and particularly by the lack of world persistence: due to memory constraints, only the player character's immediate vicinity is simulated, which means that characters and objects leave memory (and disappear forever) as soon as one turns a

[45] For this reason, I often feel like an outcast among videogame players, developers, and scholars, but I have a comrade in Gonzala Frasca: "I find most fantasy-related videogames quite boring. I have always preferred stories about human affairs and social issues to magic spells and mean dragons. This is why I always salute any attempt to bring human characters to videogames" [355].

[46] This idea of gameplay as a constructive act that produces a kind narrative is the crux of Aaron Reed's notion of *sculptural fiction* [984]. To the degree that such an act works like a collaboration between system and player, the work affording such interaction falls in the purview of Ben Samuel's *shared authorship* [1084]. Later on, I will discuss these ideas in more depth, since they relate to aspects of my framework for curationist emergent narrative. Interestingly, much of the earliest writing on interactive fiction viewed the reader as co-author [543, 48, 217, 156, 39, 1378, 1167, 416, 417] or "co-actor" [40, p. 61] or "prosumer" (portmanteau of 'producer' and 'consumer') [1119, p. 60]; these terms evoke Bertold Brecht's "spect-actor" [121].

corner. This does not reflect a design flaw, but rather a difference in emphasis. The *Grand Theft Auto* series is all about the near term: gratification is disbursed continuously as gameplay moves from visceral moment to visceral moment, and there is no reason to maintain dependency between such moments. What I personally wanted, however, was something longer term—I wanted to carry out an extended project of my own design.

A Thrilling Emergent Experience

Before putting down the game, however, I played through its linear mission structure, and it was in this unlikely context that I had perhaps the most thrilling emergent experience in all of my history with computational media. In a mission called "Psycho Killer", shown in Figure 3.2), the player is tasked with apprehending an unhinged character who seeks to kill the members of a band, Love Fist, with whom the player character is associated. The mission is designed to play out in a short car chase that culminates in the killing of the character being chased, but in my playthrough things went off the rails.

Over the course of the car chase, both of our vehicles became damaged to the point of starting on fire—in the *Grand Theft Auto* series, this indicates that the vehicle is about to explode, which means that its passengers must promptly exit or else die in the explosion. As such, both the killer and I were forced to leave our vehicles, but because this occurred in a relatively empty neighborhood near the edge of the island city, there were no nearby vehicles for either of us to hijack. Having no other choice, the killer proceeded to run from me on foot, and I chased after by the same means. This is where the situation took on an almost magical quality. After a few blocks, the killer took a turn toward the water, walked out onto a marina, and hopped into a boat that was docked there. I followed after

Figure 3.2: Scenes from the *Grand Theft Auto: Vice City* mission "Psycho Killer". Perhaps my most thrilling personal experience with emergent narrative transpired over the course of a *Grand Theft Auto: Vice City* play session in 2002. In the mission "Psycho Killer", the player is tasked with apprehending an unhinged character who seeks to kill the members of a band, Love Fist, with whom the player character is associated. The mission is designed to play out in a short car chase, but in my playthrough things went off the rails: after our cars exploded from damage incurred during the chase, the killer and I were forced to exit our vehicles and an hourlong chase on foot and by boat ensued. These images are not from my playthrough, but they depict the beginning of the mission as it is scripted.

and commandeered another boat, and from here the chase transpired on water, through the beautiful ocean passages of Vice City. By this point, I was mesmerized and had become more interested in prolonging the chase than in executing the stipulated task of killing the killer. We cruised in our boats for what felt like a long time, but eventually the killer approached a dock somewhere else in the city, disembarked, and again proceeded on foot. I did the same, and as we emerged back onto the street, the killer stole a car and I did the same. At this point I began debating whether it would be more poetic for the killer to get away—after taking such creative means, this seemed to perhaps be the just outcome—or to succumb finally to the mission's tragic design. I chose the latter, and this culmination terminated an emergent experience that I recall lasting upward of an hour.

The Personal Dimension

Why was this experience so powerful for me? Why do I still remember it fifteen years later, while I can recall essentially nothing from the game's scripted narrative? When the authored story of the mission (a brief car chase) veered off the rails into an emergent sequence that perhaps no other player ever experienced, something more akin to nonfiction obtained. This strange chase by car and foot and boat actually *happened*, and this makes it inherently more interesting—if the designers of the game had scripted a mission designed to transpire like my emergent experience did, the experience of that mission would not be as interesting.

While the actuality of the experience is important, I contend that the primary source of intrigue for this emergent sequence is something else: *I personally experienced it*. Because I was an active participant in this story, for me it does not merely work like nonfiction, but moreover like *lived experience*. Thus, my account of this emergent experience, as it has appeared on these pages, is a story of lived experience. While a particular pleasure of nonfiction is rooted in the ostensible actuality of the recounted events, an additional pleasure obtains when the recounted events were personally experienced. This is intuitive, since we encounter this phenomenon all the time. Consider the kinds of stories that we tell one another about our daily experiences—these are not inherently riveting (they would not make good novels or good screenplays), but instead are interesting because they recount actual events, and moreover lived experience. Often, such stories are more interesting, or are only interesting, to the ones who tell them, because they actually participated in the recounted events. These are the kinds of stories that are told after a day at the workplace—they may require indulgent audiences, but they are interesting enough to the teller to merit the telling.

Personal Relatedness and Story Interestingness

Here, we do not have rely completely on intuition, since a number of researchers have corroborated the argument that I am making here. Since at least the late 1970s, cognitive scientists exploring story understanding (and discourse processing, more broadly) have investigated the notion of *story interestingness*. While I discuss this work in more depth in Section 4.1.1, I will mention here that several researchers have pointed out the obvious: stories about ourselves or those we know are inherently more interesting. In a seminal paper in this tradition, Roger Schank cites "personal relatedness" as an operator that may work in tandem with other interestingness factors to boost story intrigue [1103, p. 15]. Since that paper, many others have agreed with Schank or taken the notion for granted [582, 418, 1178, 480, 972, 481, 24, 99, 1373]. In Section 3.1.2, I discussed personal relatedness at length by articulating a connection between emergent narrative and stories of lived experience. Interestingly, a recent paper by Petri Lankoski on story intrigue in videogames reports positive correlation between interestingness judgments and characteristics that are more typical of emergent narrative, namely that a game "support[s] different play styles" [643].

A Lack of Curation

While the pleasure of nonfiction obtains in emergent narrative, the pleasure of stories of lived experience obtains in the more specific case of *interactive* emergent narrative. However, as I will argue in Chapter 4 as an extension of Espen

Aarseth's critique of emergent narrative, few interactive works in this area actually produce narrative artifacts. Consider the case of my *Grand Theft Auto: Vice City* experience. The events that I have recounted did not constitute a narrative artifact merely by their transpiring, but instead the story came to exist *through* my recounting of those events. Thus, my account is a story of lived experience—one that I can look back on fondly, reaping the benefits of both nonfiction and personal narrative—while the raw experience recounted thereby is merely the subject of that story.

So while a core appeal of interactive emergent narrative is that it supports such stories of lived experience, these works tend to require that the interactor do the actual curation work to produce such stories. Typically, this curation process occurs in the mind of the interactor—or, more specifically, it may be enacted in the course of a conversational recounting to someone else—and indeed this is how stories of lived experience tend to originate. In other cases, however, curation by the interactor may result in a work of media that recounts the lived experience. In fact, this is the pattern of curation that characterizes *Oilfurnace*, the case study that appeared in the last section: for Tim Denee, the gameplay that he recounts was a personal experience, and so for him the comic is a story of lived experience. Because *Dwarf Fortress* is a *god game* [151] (the player does not control a situated avatar), Denee's sense of that experience on Dread Island being one that was lived by him may be reduced relative to my own experience in Vice City. Here, there appears to be an interesting spectrum, where certain interaction configurations are better poised to support stories of lived experience.

To return to the question of who is doing the curation: I am excited by the prospect of systems that would automatically curate gameplay to recount, to a player, her own lived experience that has transpired thereby. Aside from a particular variant of *Diol/Diel/Dial*, a project that I discuss in Chapter 8, my work that I report in this dissertation does not really venture in this direction. While both *Bad News* and *Sheldon County*—the subjects of Chapters 10 and 12, respectively—curate emergent material to form narrative artifacts, this emergent material does not (typically) result from player activity. Nonetheless, this is something that I am interested in, and a broader way of describing the task is one of *gameplay summarization*, which is an area in which I have done some work.[47]

Summarizing Interactive Emergent Narrative

On an earlier project, collaborators and I sought to summarize the emergent material produced by *Prom Week* gameplay [44]. To my knowledge, this notion of procedurally generating a post-mortem after gameplay was first explored by Anthony Davey's *Proteus* system, reported in his 1974 dissertation [241, 242], which could play tic-tac-toe and generate game summaries afterward.[48] In his 1996 masters thesis, Paul Bailey demonstrates the generation of stories summarizing tic-tac-toe, card games, and the *blocks world* [64].[49] Ian Frank also proposed the generation of gameplay postmortems in an obscure 1999 paper [353]. In our project with *Prom Week*, this meant generating narrative accounts from *playtrace data* recorded by the game. While other projects have used such *Prom Week* data for authoring support [1087, 910] or interpretation [1085], this was a first attempt at producing narrative accounts.

Along with my advisors Michael Mateas and Noah Wardrip-Fruin, I call this

[47] Of course, not all gameplay works like emergent narrative, so this category is broader than what I have described so far in this section.

[48] David McDonald's 1981 dissertation reported on an application of his system *Mumble* to the tic-tac-toe summarization task that Davey had explored [815]. In 1984, Graeme Ritchie reported on a rational reconstruction of *Proteus* [1009].

[49] For his PhD work, which he appears to have abandoned, Bailey proposed reader-centric story generation [65, 67, 66], as I will discuss later on.

task of producing narrative accounts of raw data *narrativization*. In an earlier paper, we proposed that a tool like *Playspecs* [906], which supports automated playtrace analysis, could be used to recognize interesting emergent sequences for the purposes of narrativization [1058, p. 21]. Interestingly, this process does not necessarily have to occur after the emergent sequence has terminated—indeed, the narrativization process could occur in an online manner, narrating the story of lived experience as it is unfolding. Along these lines, in his *Writing Buddy* project, Ben Samuel is currently exploring the use of *Playspecs* to recognize unfolding emergent sequences and also to consider prospective completions to them [1086, p. 393]. In another recent project, Camile Barot and collaborators have produced *Bardic*, a system that narrativizes *Dota 2* gameplay [76]. Other workers in this area have explored the summarization [685, 114, 747, 732] or narrativization of chess gameplay [389, 390], procedural commentary or narrativization for sporting events [1239, 1376, 29, 133, 669], and the narrativization of logs from other videogames [183, 181, 495, 532]. Here, the most extensive exploration has probably come in the development of procedural commentary for commercial sports videogames [998, 670, 61, 1194, 13].[50]

Personal Relatedness Without Interaction

It is worth considering how important the dimension of interactivity is to the phenomenon I have described in this section. While a feeling of lived experience will be far more activated in the case of interactive emergent narrative, it is still possible in non-interactive works. For instance, if a person encounters a story that

[50] An interesting variation on this notion comes in the form of a recent patent on technology for procedural commentary of a videogame itself, in the style of film commentary [1304]. This would take the form of a gameplay mode that procedurally triggers the playback of audio segments that provide insight about what is currently being experienced in gameplay (for example, how a level was developed).

has been generated by a computer program just for her, she may feel a personal attachment to it, since no one else has encountered it and no one else will. Thus, the kind of ephemerality that characterizes works of emergent narrative, interactive or not, may work to engender a sense that generated narrative outputs are personalized, which makes encountering such outputs a kind of personal experience. It would thus not be unreasonable to say that such cases work (to some degree) like stories of lived experience.

Computational Personal Narrative

Lastly, I would like to note that there is an interesting line of work that explores the intersection of computational methods and personal narrative. Andrew Gordon and Reid Swanson (and at times other collaborators) have developed techniques for automatically recognizing personal narratives found in blog entries, which they used to build a corpus comprising millions of examples [411, 410, 412, 408]. This corpus was then harnessed to drive Swanson's dissertation project, the aforementioned story generator *Say Anything* [1226, 1225, 1227], and others have used automatic methods to study features of these stories [1077, 967, 966, 1228, 1051, 968, 988]. Other work out of Marilyn Walker's lab (led by students Chao Hu, Stephanie Lukin, and Kevin Bowden) has demonstrated approaches for stylistic (and even multimodal) *regeneration* of such personal narratives [712, 515, 135, 714]. It would be interesting to apply these techniques to the task of narrativizing event sequences from interactive emergent narrative.

3.1.3 Analogy to Worldbuilding

Due to its emphasis on simulating the background and larger context of a storyworld, emergent narrative may be viewed as employing a kind of *worldbuilding*

[1021, 911]. In worldbuilding, an author endeavors to construct a larger world in which the characters and events of a story may be situated.

Tolkien's Aristotelian Inversion

While this is arguably an inevitable feature of all fiction [1069], in worldbuilding the world is both *preliminary* and *primary*. As the writer China Mieville states, the practice of worldbuilding curiously inverts the typical procedure of fiction authoring: "The order is reverse: the world comes first, and then, and only then, things happen—stories occur—within it" [835]. This is in a sense anti-Aristotelian, as Hanna-Riikka Roine notes in her dissertation on worldbuilding: Aristotle contends that action and plot give rise to the other fictional elements, which may be viewed as constituting the fictional world, but in worldbuilding the world gives rise to the action [1021, p. 17]. J. R. R. Tolkien, who is likely the individual most associated with the technique, called it "sub-creation": a mode of fiction crafting that entails a subprocess through which the author "makes a Secondary World which your mind can enter" [1260, p. 12]. As Roine notes [1021, p. 17], while Tolkien of course had his own influences, the mode of worldbuilding (or 'sub-creation') that he employed in his work (particular the *Lord of the Rings* series) established the fantasy genre and inspired projects outside of print fiction, including *Dungeons & Dragons* [436] (and thereby *interactive fiction* [846]) and the practice of *transmedia* [542, 253]. While worldbuilding may viewed as a reformulation of fiction's authorial process (world precedes plot), it may also be viewed as a conceptual orientation (world above plot) or a communicative orientation (world concretizes a thought experiment) [1021].

Emphasizing the Simulated Storyworld

The analogy between worldbuilding and emergent narrative is this: just as the former inverts the process of fiction authoring to proceed from world to action, the latter inverts the orientation of computational narrative such that a simulated world generates action (and not vice versa). By the interventionist approaches that characterize conventional computational narrative, namely *drama management* [654, 1326, 758, 1010] and *narrative planning* [1370, 1004], the conceptual center of gravity is the plot. Indeed, it should not be surprising then to discover that these approaches are rooted intellectually in the Aristotelian poetics that worldbuilding inverts [654, 656, 575, 758, 761, 763].[51] While this perspective may be viewed as plot-centric (Aristotelian) and others as character-centric[52] or author-centric, the brand of emergent narrative promulgated in this dissertation might be called *world-centric*.[53] While emergent narrative has typically been called character-

[51]Hartmut Koenitz has recently articulated an evocative account of the intellectual history of the drama management approach (citations inserted for clarity): "instead of emphasizing the connection to ancient drama, Mateas' and Stern's Façade can be more aptly described as based on the theoretical framework developed by Mateas himself (2001) [763], modifying Laurel's (1991) [656] reinterpretation of Neo-Aristotelian concepts by implementing Janet Murray's (1997) [870] analytical and phenomenological categories. The reference to the rather distant—several times removed—original notion is more distracting than helpful. Indeed, Aristotle's unmodified concept is inappropriate for Interactive Storytelling in its basic demand of a plot that is perfect if nothing needs to be added or can be removed, a perspective which is in fundamental disagreement with the idea of interactive narrative that needs to offer excess material for alternative paths and optional elements" [611, pp. 8–9]. As for narrative planning, its figurehead is R. Michael Young, whose earliest writings on the topic also make the Aristotelian connection [1370, 1095]. Interestingly, in his dissertation, Mark Riedl (who was a student of Young) cites Lajos Egri's *premise*-oriented process of dramatic writing [303] as the primary intellectual influence on his own approach to narrative planning [1005, p. 44].

[52]Mark Riedl [1005, p. 44] associates the character-centric orientation with the more contemporary craft advice of Lajos Egri [303] and Robert McKee [818].

[53]The tripartite model of plot-centric, character-centric, or author-centric story generation appears to originate with Paul Bailey [67, 772], whereas an earlier character vs. author duality was first articulated by Natalie Dehn in 1981 [252]. (The difference between the author-centric and plot-centric approach is that the former attempts to model the creative process of a human author, while the latter operates more directly at the level of plot manipulation.) Beginning with Masoud Yazdani's initial reporting of *Roald* in 1982 [1363], and more famously with Michael Lebowitz's first *Universe* paper of 1983 [663], practitioners have shown that systems may bridge such conceptual divides. Furthermore, others have introduced alternative approaches to these

centric [252, 1002, 702], the world-centric approach that I advocate here extends character simulation to a *massive* scale: instead of simulating the activity of characters with regard to a specific narrative premise, I am interested in simulating entire storyworlds containing hundreds or thousands of characters who live out lives that extend beyond the context of a given narrative premise. Thus, while I still view classic systems like *Tale-Spin* [822] and *FearNot!* [55] and *The Sims* [792] as exemplars of emergent narrative, I am especially interested in the kind of grandiose world simulation that is typified in *Dwarf Fortress* [17]. As such, the fundamental difference between a character-centric system and a world-centric system is the increased magnitude and granularity at which the latter simulates a larger world as a backdrop for potential emergent stories.

Thus, while the hallmark of emergent narrative is its character-centric orientation, I aim to promote an increased breadth and depth of simulation such that the result might be called world-centric. When this approach is taken, the pleasure of worldbuilding obtains, and in particular this orientation undergirds the *aesthetics of a larger context*, as I outline in Section 3.2.

---

three, such as Chris Crawford with his "verb-based" *Erasmatron* [222, p. 122] and Bailey with his reader-centric approach [65, 67, 66]. Bailey's project was proposed while he was a graduate student at the University of Edinburgh [1031, 65] and did not reach fruition, but I would like to see greater exploration of the reader-centric approach to story generation. Inspired by Bailey, Peter Mawhorter originally proposed to move in this direction with his dissertation project [783], which after some reformulation resulted in the introduction of *choice poetics*, an intellectual and technical framework that considers reader response in the context of choice-based interactive narrative [786, 769, 787]. Stacey Mason has also produced a descriptive model that incorporates reader concerns [753]. Eventually, Mawhorter operationalized his framework in a system called *Dunyazad* [785, 788, 789], which generates choice-based interactive narrative that targets particular reader responses. Along related lines, an earlier system (1998) by Boris Galitsky attempted to generate *persuasive* stories for internet ads (to persuade potential buyers) [366, 367], and Kim Binsted and Graeme Ritchie proposed generating *story puns* that target a reader response across generation instances [111, 112]. It even appears that Jim Meehan was beginning to explore reader-oriented story generation just before he left UC Irvine [824, p. 459]—unfortunately, I have not been able to obtain his paper on this new direction [825]. Finally, Pablo Gervás and Carlos León have recently argued in favor of this direction [392].

Curating the Simulated Storyworld

That being said, the presence of a massive world simulation constitutes only the first concern underpinning my framework of curationist emergent narrative. My tagline for this approach is *overgenerate and curate*, and while a world simulation satisfies the 'overgenerate' imperative, I also believe that a simulation does not on its own constitute a work of emergent narrative. In my view, it is only when the material produced by such a simulation is curated that a narrative artifact obtains. When this is done, the narrative artifact is imbued with greater intrigue because it stands against the backdrop of a larger context that is constituted in the simulated storyworld. This context on its own, however, is not narrative, and moreover it has no backdrop to stand against, since as a raw and unprocessed storyworld it *is* the entire context.[54] Thus: overgenerate *and curate*. In Chapter 5, I will return to this argument to articulate and advance my framework for curationist emergent narrative.

3.1.4 Analogy to Art Brut

I begin here with my favorite computer-generated story:

> Henry Ant was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. He was unable to call for help. He drowned. [822, p. 128]

This is one of *Tale-Spin*'s famous *mis-spun tales* [822, p. 126], a collection of outputs reported by its creator Jim Meehan as illustrative examples of the story

[54]To return to Tolkien, we might characterize the negative reception of *The Silmarillion* (as opposed to *The Hobbit*) [347, 14] as being rooted in the failure of the former to curate a distinct narrative artifact from the congeries of the storyworld's raw history. Writing about *Dwarf Fortress*'s Legends mode [279], Stephanie Boluk and Patrick LeMieux make the same connection: "The result is a textual archive that reads more like *The Silmarillion* than *Lord of the Rings*" [126, p. 143]. This analogy does not perfectly apply, however, since the book is still a narrative account, not literally a raw storyworld (in the sense that a computer simulation may be one).

THE POPOVICHES LIVE IN A DISTANT PROVINCE.
THE FATHER IS VLADIMIR.
THE MOTHER IS DUNIA.
THE OLDEST SON IS BORIS.
THE YOUNGER SON IS EREMA.
THE YOUNGEST SON IS FOMA.
VASILISA IS THE ONLY DAUGHTER.
EREMA ASKS DUNIA WHERE IS YOUR WISDOM.
DUNIA SAYS THAT MY WISDOM IS IN A MAGIC EGG.
DUNIA MURDERS EREMA.
A LAMENT IS SUNG FOR EREMA.
EREMA IS WANDERING.

EREMA MEETS A STOVE ALONG THE WAY.
THE STOVE PROPOSES THAT EREMA EAT THE MEAL WHICH IT HAS PREPARED.
EREMA RESPONDS BY PARTAKING OF THE MEAL.
A MAGIC POTION IS CONSUMED BY EREMA.
EREMA OBTAINS INCREDIBLE SPEED.
EREMA STARTS BACK HOME.
EREMA RETURNS HOME.

"Henry Ant was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. He was unable to call for help. He drowned."

Figure 3.3: Minneapolis artist Dietrich Sieling's *Shirt Off Geoff* (2014) juxtaposed with stories produced by Sheldon Klein's folktale generator and Jim Meehan's *Tale-Spin*. Sieling, who is one of my favorite contemporary visual artists, has a striking idiosyncratic style that works independently from entrenched artistic traditions. What would an unabashed idiosyncratic computer-generated aesthetic for story generation look like? Perhaps Klein's surreal output and Meehan's "mis-spun tale" shown here are indications. I would like to see experimentation in emergent narrative by which the unavoidable quirks of computer simulation are celebrated and emphasized, rather than shrouded and concealed.

generator going awry (usually due to gaps in Meehan's knowledge engineering).[55] When things went "right", the resulting stories were regenerated Aesopian fables with dubious prose. I am not alone in appraising the mis-spun tales as far more evocative than their correctly spun counterparts. Indeed, Noah Wardrip-Fruin has noted that these curious artifacts are the system's most famous (and most frequently reprinted) outputs [1311, p. 130].[56] Wardrip-Fruin cites the "amusement value" of the tales, and others might characterize them as a kind of *glitch art* [832], but I think there is something more here.

Certainly the story above expresses a kind of glitch (though one rooted in human authoring), and I do agree that the story is humorous—but its comedy

[55] An aesthetically kindred example, produced by a story generator based on *Tale-Spin*, appeared on the `comp.ai` newsgroup in 1993: "Once upon a time there was a bear named Jack. Jack was dead. Jack was still dead. The end." [469, n.p.].

[56] As Wardrip-Fruin explains, this may be due in part to the prose of the mis-spun tales being hand-authored by Meehan (from the underlying computational representation), whereas the prose of the successful outputs was computer-generated.

is dark and tragical and strange and alien. I contend that there is an *uncanny* quality of the tale that yields its peculiar aesthetic impact. It does not work like a human story—that is, it does not seem to be a story that a human would be compelled to tell. Below, in Section 4.1.1, I discuss the notion of *tellability*: a narratological (and sociolinguistic) concept that pertains to whether a story is even tellable (where the stakes are losing face due to the wasting of a recipient's time and cognition). Put differently, a story is a recounting of a tellable sequence of events. It is tempting to say that this story is intriguing because in human culture it is not tellable, but in the mind of this computer, it is. I do not think this is the case, though: it is easy to generate untellable stories that are not intriguing, but such tales are simply *boring*. This story is not boring—there is something strangely enchanting about it—and in this section I intend to characterize this mysterious quality so that it may be explicitly targeted (or at least celebrated) in future generative works.

A Computer Art Brut

I view this mis-spun tale and related examples as representing a kind of computer *art brut*. Sometimes called 'low art' or 'folk art' or 'naïve art' or 'outsider art',[57] art brut (literally 'raw art') is a term that was coined by artist and thinker Jean Dubuffet to refer to artistic practices that exist outside of the art establishment and that are (usually incidentally) in opposition to its aesthetic hegemony [927]. This art—"unscathed by artist culture" [274, p. 31]—has often been identified with children [160], mental illness [720], and developing nations [839], but such factors are not necessary or sufficient conditions for art brut, but rather

[57] I agree with others who have deemed 'outsider art' a poor synonym, since that phrase is often used to refer to a broader category of works [722, 190].

recurring circumstances.[58] While art of various kinds may evoke the 'art brut' classification, certain aesthetic qualities underpin the idea: a sense of the crude, uncanny, alien, eccentric, deranged, marginalized, pure [274, 1335, 1323, 839]. As such, the form is bound primarily by a set of *family resemblances* [1345], rather than any concrete concerns pertaining to medium, content, and so forth.

I am interested in asking *how would a computer art brut work?*, and I think the uncanny mis-spun tale reported above is a potential answer. It is art by a computer that is distinctly removed from the aesthetic sensibilities of human art. Here is another generated story that I love for the same reason:

> The Paranovs live in a distant province. The father is Emelya. The mother is Maria. The oldest son is Foma. The younger son is Baldak. Vasilisa is the only daughter. Emelya says Baldak, do not go to the house. Emelya dies horribly. Baldak goes to the house. A bear flys [sic] into the distant province. The bear asks Marco where is Baldak. Marco says that Baldak is in the house. The bear orders the murder of Baldak. Baldak is secretly freed. Baldak is wandering. Baldek meets a forest knight along the way. The forest knight brawls in a forest hut with Baldak. The forest knight twice repels Baldak. They fight for the third time. Baldak defeats the forest knight. A magic carpet is given to Baldak. Baldak starts back home. Baldak returns home. [595, p. 64]

This eerie tale was produced by a system developed in the mid-1970s by Sheldon Klein and his students to operationalize Vladimir Propp's famous model of Russian folk tales [960].[59] In an improbably *more* surreal output, one character keeps acting even after death (though perhaps in a liminal hereafter):

[58] In attempting to celebrate the aesthetic sensibilities of artists who are deemed to be 'outside' of some boundary, there is clearly a risk that such demarcation will result in an othering, or in a patronizing fetishization, of the characteristics that are viewed as contributing to one's outsider status. This has been articulated before [1347, 354, 320], and though the stakes are different when considering the otherness of a computer, I think these considerations are still worth acknowledging.

[59] As Klein explained, he did not set out to create strange stories, and in fact he viewed deficiencies in the generated outputs as proof that Propp's model was itself deficient (citations inserted for clarity): "That the results are bad Russian fairytales is a major theoretical significance, for they confirm that Propp's model is for the syntax of fairytales, and that there must also exist a higher level semantics of fairytales outside the description of Propp. Our experiment

> The Popoviches live in a distant province. The father is Vladimir. The mother is Dunia. The oldest is Boris. The younger son is Erema. The youngest son is Foma. Vasilisa is the only daughter. Erema asked Dunia where is your wisdom. Dunia says that my wisdom is in a magic egg. Dunia murders Erema. A lament is sung for Erema. Erema is wandering. Erema meets a stove along the way. The stove proposes that Erema eat the meal which it has prepared. Erema responds by partaking of the meal. A magic potion is consumed by Erema. Erema obtains incredible speed. Erema starts back home. Erema returns home. [595, p. 65]

The evocative *computer art brut* aesthetic of these generated tales is further revealed in their original form, as presented in a technical report via printout by a drum printer, which produces a distinctly computer-generated look; this is shown in Figure 3.3.

Uncanny Emergent Narrative

While Klein's system is not an example of emergent narrative, *Tale-Spin* is, and in any event we might consider how projects taking this approach could work like art brut to yield a striking idiosyncratic aesthetics. Due to the nature of emergent narrative, such qualities would likely be based in the peculiarities of computer simulation—perhaps the gaps or exaggerations or other alien qualities of a simulated storyworld could be emphasized to evocative effect. Such qualities are probably unavoidable in computer simulation, but I would like to see projects in emergent narrative that embrace them. The stories recounted in this way would be openly and proudly about the surreal ontology of the storyworlds from which

provided a concrete verification of the points about Propp made by Levi-Strauss (1960) [684] and Meletinski *et al*, 1971 [827])" [600, p. 2]. I am, however, interested in celebrating these "deficiencies", especially when they are rooted in the alien peculiarities of a distinctly computational sensibility. Note that this is different from celebrating the mere bad. An earlier project in Proppian story generation that produced more mundane stories was abandoned because its practitioner, Joseph E. Grimes, found the outputs to be boring [1039]. I do not advocate boringness, and that is exactly why I am not intrigued by *Tale-Spin*'s "successful" outputs.

they emerge—weird tiny abstract worlds that work. Again, this does not mean celebrating what is merely bad, but rather what is surreal and striking. I return to this idea in Section 3.2, and in Section 4.2.6, I criticize Meehan's approach to *Tale-Spin* for its posturing toward a boring human aesthetic. Of course, these are highly subjective considerations, and in this section and the others I am certainly proceeding from my own personal inclinations. Nonetheless, I think it is worth at least *considering* what an unabashed *computer art brut* might look like, and for our purposes here, an alien emergent narrative.

3.2 The Aesthetics of Emergent Narrative

Through its resemblances to the various forms outlined above, emergent narrative unlocks some of the pleasures that are associated with them. To conclude this chapter, I will briefly identify a set of such appeals, which are also constituted in additional peculiar features of the form—I view these pleasures as constituting an *aesthetics of emergent narrative*. Note that my aim here is to provide a brief first articulation, which makes a deeper engagement of these ideas in terms of existing aesthetic theories the object of future work.

3.2.1 Aesthetics of the Actual

Due to its resemblance to nonfiction, articulated in Section 3.1.1, emergent narrative unlocks the hallmark pleasure of that form, which may be referred to as the *aesthetics of the actual*: when a story recounts (ostensibly) actual events, it will likely be more interesting than a fictional, but otherwise identical, counterpart. This is the basis for the marketing trope characterized by the phrase "based on true events", and it is a pleasure that undergirds enjoyment of emergent narrative.

3.2.2 Aesthetics of the Personal

Through its connection to stories of lived experience, identified in Section 3.1.2, (interactive) emergent narrative yields an *aesthetics of the personal*: when a story recounts events to which a recipient is personally connected, it will likely be more interesting to her (with all other things being equal). As I noted in my discussion above, this is primarily a feature of interactive works of emergent narrative.

3.2.3 Aesthetics of a Larger Context

By an analogy to worldbuilding, as discussed in Section 3.1.3, emergent narrative is bolstered by the *aesthetics of a larger context*: a special intrigue is attached to stories that transpire against the backdrop of a larger storyworld. While all narrative works imply (and at times concretely model) a storyworld, world simulation is the organizing principal in emergent narrative.

3.2.4 Aesthetics of the Uncanny

As a kind of simulationist art brut, following my argument in Section 3.1.4, emergent narrative produces a unique *aesthetics of the uncanny*. As I discussed above, there is a (subjective) gray area between the uncanny and the mere bad, but the tiny abstract storyworlds that are simulated in emergent narrative seem to buttress an uncanniness that does not manifest in other forms of (procedural) narrative. Stephanie Boluk and Patrick LeMieux have celebrated this feature in one of the form's major works: "*Dwarf Fortress* embraces the granularity of computation over the specularity of graphical realism, delighting in the strange, formal logics of the computer" [126, p. 126]. In a related argument, Ian Horswill has called for deeper consideration of genre and tone in computational media, since some orientations do not seem to work procedurally given the state of the art in available technologies [504]. As a specific example, he argues for the targeting of *farce*. Likewise, Tanya X. Short has recently advised game designers to "lean into the comedy" and embrace the peculiar dynamics of procedural characters [1150, p. 114], a stance echoed in Kenneth Chen and Stefan Rank's recent study of humor in interactive emergent narrative [179]. This kind of coming to terms with the broad aesthetic potentials of procedure is what I am getting at with my evangelism of the uncanny.

3.2.5 Aesthetics of the Unauthored

As a form of procedural generation, and moreover the least authorial approach to story generation, emergent narrative is characterized by an *aesthetics of the unauthored*: there is a peculiar delight in beholding an artifact that was not created by a human. This is a phrase that I borrow from Jason Grinblat, who in a Twitter thread on generated world maps remarked, "I love procedural generation and the aesthetics of the unauthored" [426, n.p.].

3.2.6 Aesthetics of the Coauthored

In the case of interactive emergent narrative, a player may have the sense of sharing in the authorship of an ongoing experience—this supports an *aesthetics of the coauthored*. Ben Samuel's recent dissertation explores this notion at depth through his framework for *shared authorship*, which concerns patterns of interaction whereby a human and machine collaborate to compose a story that neither entity could have composed on its own [1084]. Related to this framework is Aaron Reed's notion of *sculptural fiction* [982, 984], in which a player composes a story graph, instead of traversing one.

Due to its bottom-up workings, player interactions are more easily incorporated into emergent narrative than other forms of procedural narrative: in emergent narrative, the storyworld evolves in response to player actions according to the rules that govern its world simulation, whereas by interventionist approaches an experience manager must take action to *repair* the storyworld in response to player actions (e.g., [1000, 687, 919, 1372, 1012]). As I explained in Chapter 2, this was the impetus for Ruth Aylett's influential exploration of emergent narrative. Because emergent narrative (in interactive works) depends more on bottom-up player actions than top-down story policies, shared authorship is more potent in the form, and the aesthetics of the coauthored are in turn amplified.

3.2.7 Aesthetics of the Uncovered

In his feedback on an earlier draft of this thesis, reading-committee member Jonathan Lessard identified an additional feature of emergent narrative that is related to, but distinct from, the aesthetics of the coauthored:

> I really like the parallel you make between the work of the historian and the relationship to simulated storyworlds. To me, this should be pushed even further. In your list of aesthetics, I would add: "the aesthetic of the uncovered" [which] is perhaps the same or close to your aesthetic of the coauthored. It's about the pleasure of discovering/unearthing/excavating/witnessing an interesting story *yourself*. This is the historian's pleasure when finding something surprising when going through their archive. This is *your* pleasure when you find these superb nuggets from your simulations, and, as you become narrator, it becomes our pleasure as you know your audience and know what will be interesting to us.[60]

This is indeed a core aspect of the aesthetics of emergent narrative, and one that ties in nicely with the other patterns that I identify in this section. In this thesis, I argue at length that, with regard to computer simulation, interesting stories can only obtain through an act of curation. When the curator (or her

[60]Personal communication, July 27, 2018.

audience) encounters such a story, her expended curatorial effort is exchanged for added aesthetic value, because the story has been *uncovered*—what would have otherwise gone unnoticed is now beheld, and this produces a distinct pleasure.

3.2.8 Aesthetics of the Improbable

Part of the intrigue of true stories is that they are inherently *improbable*: coincidental real-world event sequences generally lack narrative intrigue, or even narrative structure, and as such it is a notable occurrence when something interesting actually does happen in the real world. This is the *aesthetics of the improbable*. In contrast to this, the very project of fiction writing is to come up with interesting events, and as a result all works of fiction tend to be 'interesting' with regard to basic notions of narrative intrigue. But when every story is interesting, interestingness itself is highly probable, and so the bar for intrigue is raised—this process may culminate in what Guy Bergstrom has called the *Michael Bay school of storytelling*:

> People want a thrilling ride? The Michael Bay School of Storytelling says OK, let's blow their minds with the most intense story ever. Except when everything is dialed up to 11, the audience goes numb. [101, n.p.]

This pitfall can also characterize works of procedural narrative in which top-down methods, driven by experience managers or other mechanisms of artifice, are employed in an effort to always produce interesting outputs. Again, when every story is interesting, that quality becomes suspect, and paradoxically the stories become boring. In emergent narrative, however, interesting event sequences are improbable, as in the real world, and so stories that do actually emerge tend to be inherently more interesting. As such, the aesthetics of the improbable is present in emergent narrative.[61] In a nice example of this phenomenon, Stephanie Boluk and Patrick LeMieux describe how *Dwarf Fortress* players have celebrated the narrative quality of the improbable:

> Another example of translation can be seen in the painstaking research and attempts to narrativize the remarkable history of the dwarf Tholtig Cryptbrain, a statistical anomaly discovered in *Dwarf Fortress*'s Legends mode and the sole survivor of a century-and-a-half-long war in which she claimed 2,341 unique kills. Tholtig's story is a death-defying feat not of skill but of chance and probability. [126, p. 135]

As I explain in Chapter 5, curationist emergent narrative (the subject of this dissertation) is about automatically identifying interesting stories that emerge from simulations. A system that can curate a simulated storyworld in this way may produce interesting story after interesting story, but critically each one has improbably emerged from a simulation. When this improbability is expressed in the media experience that houses the curated stories, the aesthetics of the improbable are maintained. Thus, by relying on curation instead of artifice or intervention, interestingness and improbability are no longer incompatible, but are instead complementary.

[61] Regarding the Bergstrom quotation above, Ian Horswill imparted the following in his feedback on an earlier draft of this thesis: "I don't think this is arguing what you want it to. You want to argue about improbability but I think this is arguing more about temporal dynamics—you have to have variations in intensity, within a given story, because humans are essentially sensitive to changes more than to absolute values. You see that in all time-based media—authors are always scheduling intensity and variation, putting in times to let the audience rest, foreshadowing and building up to an event to let the audience anticipate it, for example" (personal communication, July 28, 2018). This is a great point, and indeed reasoning about such temporal dynamics feature heavily in the computational systems that drive relatively top-down approaches to procedural narrative (for instance, in *Façade* the fundamental unit of narrative is the *dramatic beat* [776]). What I am attempting to write about here is dynamics at the level of a system's *possibility space*: what are the intensity dynamics across the instances in this space? In this case, the instances are event sequences, and in this dissertation I am promulgating an approach whereby few instances in a system's possibility space are compelling. This may seem paradoxical, from the standpoint of media creation, but this is what makes true stories interesting in the real world—they are improbable with regard to the larger space of possible actual event sequences—and that is the kind of pleasure that characterizes emergent narrative.

3.2.9 Aesthetics of the Vast

Emergent narrative is vast: through the mechanism of massive rich simulation, its storyworlds have considerable depth and breadth, and a higher-order vastness obtains with regard to the possibility space constituted by all a system's generable storyworlds. By centering on the world, and not the plot, emergent narrative overwhelms with its sheer accumulations of material. This can be overwhelming in a good way (*Dwarf Fortress* players love its overwhelming complexity and detail), but also in a bad way (*Tale-Spin*'s extraneous prose may obfuscate narrative intrigue). When vastness contributes to the success of a work, the *aesthetics of the vast* is at play. As Noah Wardrip-Fruin noted in his feedback on an earlier draft of this thesis, my articulation here evokes the concept of the *technological sublime* [215, 885]. Indeed, others have made the connection between this notion and computational media (and sometimes procedural generation more specifically) [125, 35, 1145, 105]. As I have already noted, a deeper engagement of these ideas in terms of existing aesthetic theories remains for future work.

3.2.10 Aesthetics of the Ephemeral

Because its stories are constituted in improbable event sequences occurring in simulated storyworlds, there is an appealing ephemeral quality to emergent narrative—this might be referred to as the *aesthetics of the ephemeral*. In beholding an emergent story, a reader (or player) can be assured that no one else will encounter this artifact: it has never existed before and will never exist again; in a sense, it is *just for her*. This sensation is especially present in projects that feature procedurally generated storyworlds, since the actual worlds, and all their characters, are likewise ephemeral.

Chapter 4

The Pain of Emergent Narrative

The essential critique of emergent narrative is that stories produced by the method are often bad, or moreover, are not even stories at all. In this chapter, I unpack this critique to discuss a series of painful shortcomings that have plagued emergent narrative since its inception in the middle of the last century. As I will argue in the next chapter, none of these points represent inherent limitations to the approach, as some have suggested, but rather shortcomings in a number of its notable works. As in the last chapter, I will refrain from discussing my own practice here, but I will note that in Part II, I critique my projects (sometimes brutally) with respect to the issues that I identify here. I hope that this chapter provides insights that enable not only improved crafting of emergent narrative, but also refined interpretation of works in the area—more specifically, particular misunderstood efforts that are of historical importance. Finally, while my primary aim here is to identify a series of pitfalls that have worked to hamper the form, in each case I will outline potential solutions, many of which have quietly been implemented in existing works. As such, this chapter also loosely suggests a *poetics of emergent narrative*.

4.1 Simulation Pains

Since at least Natalie Dehn's 1981 paper "Story Generation After *Tale-Spin*" [252], practitioners have identified a fundamental pitfall to the simulationist approach taken in emergent narrative: when characters are given free reign, the simple result of running a given simulation instance may not be a good story. As Dehn put it, "Setting up an initial story world such that things work out in an interesting way is, at the very least, tricky business" [252, p. 16]. Indeed, even in her seminal first paper on the topic, Ruth Aylett noted, "one of the risks of emergent narrative is that it may not emerge—the unpredictability that makes it interesting also makes it in some sense fragile" [50, p. 85]. Thus, works of emergent narrative may be doomed before the narrative ever makes it to the surface.

In this section, I identify and discuss a series of issues that pertain to the crafting of the underlying simulation that models and evolves the storyworld out of which narrative may emerge. These issues may be viewed as design pitfalls that simulation authors should take into consideration when developing their own systems. Some of these problems are germane to the broader area of story generation—or, even more generally, to procedural generation and computer simulation—while others seem to be endemic to emergent narrative.

4.1.1 Boringness

A fundamental pitfall in emergent narrative pertains to the level of interestingness inherent in the simulation domain—that is, some simulations are too *boring*. In this section, I will explore this idea by touching upon scholarly investigation of the notions of *narrative criteria*, *story interestingness* and *tellability*.

Criteria for Narrative

In the most extreme cases, a simulation may be so lacking in the intrigue that we demand of stories that it cannot generate any good stories, or even anything that could be called narrative at all. While this critique has been levied against systems like Jim Meehan's *Tale-Spin* [822], an actual extreme case would be something like the famous *blocks world* of Terry Winograd's *SHRDLU* system [1343]. Many projects have used this microworld as a testbed for planning [1166], and in such projects the trace of a path to a goal state works just like story generation works in *Tale-Spin*—the generated output is a *simulation trace*—but the resulting traces would never be called stories. This is intuitive, but still curious, because such a project satisfies the basic *narrative criteria* implied by Meehan's design of *Tale-Spin*: a story occurs when an entity has a goal and executes a plan that terminates in either the goal state or a failure state.[1]

So what is missing? The most glaring omission is characters—let us say then that the blocks world and other microworlds without characters (a lunar base [1356], a fueling station [472], a power circuit [145], etc.) are probably not capable of producing narrative. Let us then consider one that does feature characters: the *clowns world* of Robert F. Simmons, which concerns a clown who can stand on his head, do tricks with a pole, and sail to a lighthouse on a boat [1156]; Figure 4.1 shows the system in action. Is this storyworld capable of producing narrative? The issue is murkier, but conveniently in this case we can say it cannot because the clown does not pursue goals beyond executing individual tricks that are specified by a human user. A later microworld called *Simon City* reintroduces the confusion, however: it is the home of a simulated taxi driver named *Elmer*,

[1]Several researchers around this time identified problem solving as a critical feature of narrative [1033, 115]. Robert Wilensky noted that obstacles to solving such problems are equally important [1340, p. 354]. The importance of an obstacle is taken for granted in the remainder of this section.

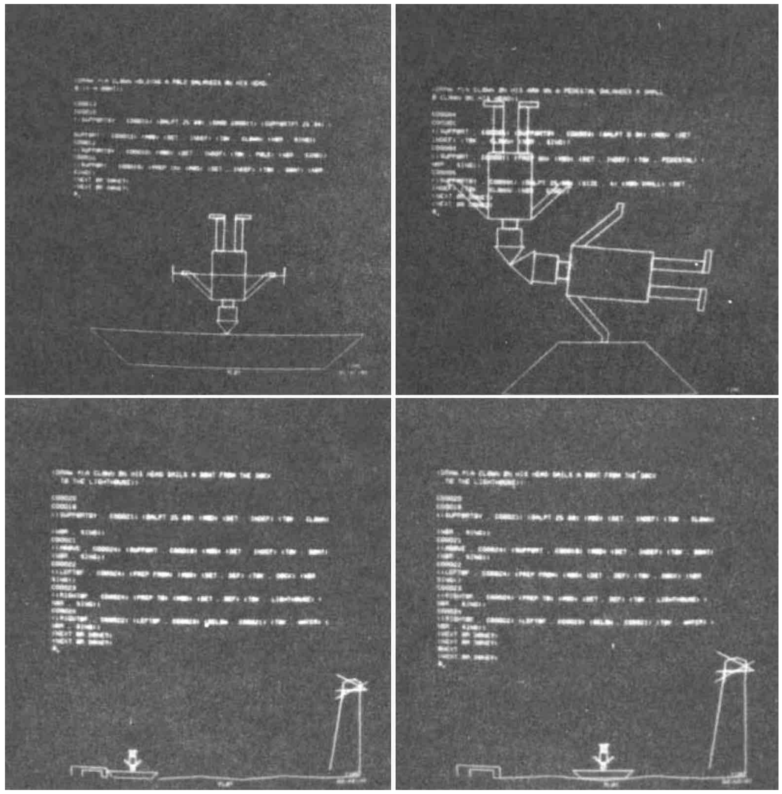


Figure 4.1: A simulated clown does tricks and attempts to sail across a bay in Robert F. Simmons's *clowns world* (1975). This delightful *microworld* concerns a clown who can stand on his head, do tricks with a pole, and sail to a lighthouse on a boat. The system features a character who pursues goals and overcomes obstacles, but do actual stories obtain thereby? The clowns world seems to reside right at the threshold of narrative potential, but I contend that its simulated events are not quite interesting enough to constitute (good) stories.

who executes plans in order to drive around the city [820].[2] This is a virtual character who has goals and executes plans in order to overcome the obstacles to achieving those goals—there is even a notion of other characters who stand in the way: modeled drivers that may inhibit *Elmer*'s driving. Is a trace through *this* simulation a narrative? It seems like it could be—and it is certainly closer to being one than the blocks world and the clowns world—but certain features still

[2]Beautifully, the city is named for AI pioneer Herb Simon, and its streets are named for other well-known figures in the field: Winograd Way, Schank Strip, the Minsky–McCarthy Freeway. I find the early microworlds to be quite evocative when they are viewed as aesthetic objects—a kind of simulationist art brut, in line with my argument in Section 3.1.4.

seem to be missing.

Let us return to Meehan's conception of *Tale-Spin*, which he articulates, in the abstract to a conference paper on the system, in terms of a more extensive set of narrative criteria :

> Tale-Spin is a program that writes stories by using knowledge about problem solving, physical space, interpersonal relationships, character traits, bodily needs, story structure, and English. [823, p. 91]

Leaving aside both 'English' (since our purview here is an underlying simulation) and the teleological feature 'story structure'—this is useless to us here, since we are trying to figure out what exactly that means—perhaps we can say that what is missing in *Simon City* is interpersonal relationships, character traits, and bodily needs. Here, then, is a computer-generated paragraph that has confounded me in my project of uncovering the forgotten early history of story generation [1039]:

> Mary was wrestling with a bottle at the liquor bar before John helped her with it. John saw Mary wresting with a bottle at the liquor bar. John went over to help Mary with a bottle before he drew a bottle cork. John and Mary together drank the champagne. [1155, p. 903]

The paragraph was generated by a project by Robert F. Simmons and Jonathan Slocum in 1972. Simmons also created the *clowns world*, mentioned above, and he was Sheldon Klein's collaborator on some of the earliest work in natural language generation [1158, 604, 603, 1159]. In fact, Klein's later work in story generation [601, 597, 595] evolved out of this collaboration, so we can say that those projects and the one at hand are cousins. While Simmons and Slocum were not interested in story generation,[3] this paragraph is reminscient of the prose of Klein's generated murder mysteries [597], discussed at length below. But while Klein's outputs are

[3]Later on, toward the end of 1970s, Simmons and his student Alfred Correira would produce *Telltale*, one of the major early story generators [1157, 214].

clearly stories, this paragraph does not seem to be as narrative.[4]

Why is this? Linking back to Meehan's story criteria, we can say that the paragraph pertains to problem solving (how to open the bottle), physical space (the bar), interpersonal relationships (John vis-à-vis Mary), character traits (John seems helpful), and arguably even bodily needs (perhaps they each went to the bar as a way to meet someone). So why is this not quite a story? It has all of Meehan's components, and it also has explicit causality, which in Section 4.1.6 I will argue is a necessary condition for narrative. Perhaps the framing of Simmons and Slocum is central—after all, they do not say it is a story.[5] While this is probably a contributing factor, I think the answer is a different but equally simplistic one.

[4]In a related example, Robert Wilensky argues that the famous Schankian "restaurant story" [1105, p. 153] is in fact not a story at all [1340, p. 346]. As I show momentarily, the example in question here is in fact more challenging to account for, since it would actually still meet Wilensky's revised criteria for a story (problems with obstacles, as discussed below).

[5]This issue of framing is very interesting to me, since it seems to be the heuristic with which scholars delineate the tradition of computational art from concurrent developments in other modes of computational work. Here, a prime example pertains to two of the earliest text generators, both of which emerged out of early American work in that area. The first of these, *Auto-Beantik*, was framed as an effort in computer poetry [1357, 1358], and it has been remembered as a computational art project accordingly [877, 458, 365]. Contemporaneously with that project, Victor Yngve utilized essentially the same method (context-free grammar) to produce syntactically well-formed nonsense sentences, partly in an effort to test a theory of English syntax [1368]. Yngve's system has not been remembered as computer poetry, but instead it shows up in the (scant) histories of natural language generation, where intriguingly early efforts in computer poetry do not appear [734, 819, 565, 80]. (Fascinatingly, however, one strangely enchanting sentence generated by Yngve's system would charm a number of poets: "What does she put four whistles beside heated rugs for?" [836, pp. 4–5][948][1181, p. 129][1288][957, p. 43].) At the time, the systems were viewed as the two prominent examples of text generation [383, p. 387], but eventually they would be remembered according to how their creators respectively framed them. I am interested in viewing all scientific projects as artistic endeavors and all artistic endeavors as scientific projects—by doing this, we reveal a new series of evocative aesthetic objects and a more robust history of human inquiry. Indeed, in my digging I have found that most of the very earliest advances in natural language generation by computer—from Strachey's love-letter generator [1205, 1312] to early computer poets both known [365] and unknown (which I hope to report on soon)—were computational art projects. It is both exciting and disheartening to imagine how some of the early AI efforts could have instead been monumental mid-century artworks. For example, I think Robert Abelson's *Goldwater Machine* [9, 8, 1311] instantiated a fundamentally new kind of representational artistic method. If it had instead been framed as an aesthetic contribution, I think it would be remembered today as a touchstone artwork, and perhaps even the progenitor of a distinct simulationist movement.

Story Interestingness

Some simulation domains are so boring that they seem to be fundamentally incapable of producing narrative, or at least anything that would be considered compelling narrative. So what makes a domain interesting? To be clear, by domain I refer generally to the stuff of the simulation: everything that it models, including setting, characters, attributes, actions, and so forth. To me, it seems that the most critical issue is the kinds of actions that are possible given what has been authored for the domain model. *Simon City*, for example, models a domain that I find inherently interesting (cities), but its sole action type (*Elmer* the taxi driver driving around) lacks intrigue.

Here, we can turn to the rather substantial body of work on *story interestingness*, which was a popular topic of research among cognitive scientists in the 1980s [1103, 582, 1341, 140], and one that continues to receive attention today through work by researchers such as my colleague Morteza Behrooz [94, 95]. A seminal paper in this area is Roger Schank's "Interestingness: Controlling Inferences" (1979), in which he posits three mechanisms that drive story interestingness [1103]. The first of these is an operator that he calls *unexpected events*: stories in which things happen unexpectedly for a given listener may be interesting to that listener. Next, he identifies another operator, *personal relatedness*, which works similarly: a story about oneself or someone one knows may be inherently interesting. Note, however, that Schank identifies these mechanisms not as being categorical in function, but as *operators* that take the single argument of a topic. Specifically, Schank identifies (as a third mechanism for interestingness) a set of topics that he calls *absolute interests*: danger, death, power, sex, money, destruction, chaos, romance, disease, "and many other concepts and issues of this type" [1103, p. 15]. As such, the larger idea is that stories may be interesting for pertaining to the topics of absolute interest, and they may be especially interesting if such topics are amplified by the operation of unexpectedness and or personal relatedness: “If someone you knew died from having sex for a lot of money, that would be very interesting by my rules” [1103, p. 16].

As Behrooz notes, Schank’s idea of absolute interests was supported in articulations of later related concepts including Robert Wilensky’s “human dramatic situations” [1340, 1341], Peter Freebody and Richard C. Anderson’s “generically important elements” [357, p. 32], and Behrooz’s own “instinctive interests” [95, p. 3]. Others have referred to this as *situational interest* [615, 478, 479]. Unpacking Schank’s notion of unexpectedness, Wilensky notes that storytelling is predicated on generating expectations (in the mind of the recipient) as to what comes next [1340, 1341], which makes story understanding a kind of problem solving [115]. In this regard, Walter Kintsch identifies a specific pattern by which particular assumptions are evoked (or even elicited) in the mind of a listener as a clever set-up for violating those assumptions later on [582]. George Mandler posited a similar theory a few years later [733], and other researchers have since corroborated this idea through experimental investigation [361, 157]. Additionally, Bruce K. Britton and Asghar Iran-Nejad independently key in on the importance of the listener eventually arriving at a state of resolution with regard to the violation of her assumptions [140, 526], through what Sung-il Kim identifies as an act of inference [581]. Generally, researchers in this area are interested in the way that stories generate interest by stimulating cognitive procedures in the mind of a recipient—this is what Kintsch calls *cognitive interest* [582, p. 88]. Notably, this phenomenon connects also to cognitivist narratology and narrative psychology, where narrative is considered as a mechanism for mental simulation [316, 1385, 887, 743, 130].[6]

[6]Intriguingly, inasmuch as the act of *crafting* a story is viewed as being based in a kind of mental simulation, the peculiar model of creative writing implicit in early story generators like

As part of their ongoing project, Behrooz and his collaborators provide an additional set of useful interestingness factors: *common themes*, such as 'being an underdog, growing up poor, being bullied in school"; idiosyncratic *topic interests*, examples of which might include "geography, sci-fi movies, fireworks"; a variant of Schank's personal relatedness that pertains more specifically to a kind of *reminiscence* about one's past; and the *implicit familiarity* of subject matter that evokes a visceral sense of familiarity or recollection in the recipient, such as a red rose [95, p. 3]. Finally, Selmer Bringsjord and Dave Ferrucci make use of a notion similar to Behrooz's common themes, which enables an "engineering trick" that they utilize in their story generator *Brutus*:

> One of our tricks for ensuring (at least a degree of) interestingness is this: Formalize immemorial themes present in belletristic fiction, and have Brutus generate stories in accordance with this formalization. [138, p. 6]

Tellability

Lastly, before connecting these ideas back to our discussion of simulation concerns, let us consider a body of work that concerns the more fundamental issue of whether a story is even *tellable* in the first place. This area originates in the work of sociolinguist William Labov (and collaborators), who identified the notion of a *reportable event* [635] as one whose reporting (in a narrative) is socially merited:

> telling a narrative requires a person to occupy more social space than in other conversational exchanges—to hold the floor longer, and the narrative must carry enough interest for the audience to justify this action. Otherwise, an implicit or explicit "So what?" is in order [637]

Defined more concisely, "A reportable event is one which justifies the automatic reassignment of speaker role to the narrator" [631, p. 405], and indeed Harvey

Saga II [859] and *Tale-Spin*—that an author comes up with a story by simulating a world in her mind (see Section 4.2.2)—is supported decades later by social science.

Sacks emphasizes that this process requires ongoing maintanance [1076]. Labov notes that certain kinds of events are always reportable—for instance, death and injury [635, p. 301]—an idea that corresponds to Schank's absolute interests, Wilensky's human dramatic situations, and other kindred notions. More broadly, a reportable event is one that is new (and newsworthy) or unexpected [1076, 637, 882]. Neal Norrick lists some additional sufficient criteria: "uniqueness and/or humor" and "the prospect of conarration" [882, p. 323].[7] Interestingly, Labov describes the construction of oral narrative as a causal backchaining from a "most reportable event" to the terminus of an event that does not require reporting:

> Before a narrative can be constructed, it must be pre-constructed by a cognitive process that begins with a decision that a given event is reportable. Pre-construction begins with this most reportable event and proceeds backwards in time to locate events that are linked causally each to the following one, a recursive process that ends with the location of the unreportable event—one that is not reportable in itself and needs no explanation. [632, p. 37][8]

A more general term for Labov's 'reportability' is *tellability*, which may be conceived as pertaining not strictly to the events of a story, but more broadly to its *stuff*. As Wolf Schmid notes, tellability may actually be predicated on the absence of expected events [1115, p. 13]. Indeed, for Marie-Laure Ryan, tellability is all about the suggestion of actions that do not occur (more precisely, that occur in plausible alternative possible worlds):

> Why is it that deception makes better tales than honesty, error than insight, broken promises than fulfilled obligations, temporary failure than continued success? Because all the above situations generate unrealized projections, and weave several parallel stories into the thread of the historical events. [1064, p. 138]

[7]Following Elinor Ochs and Lisa Capps [891], Norrick defines a parabolic tellability, with boringness on one side and taboo (vulgarity, oversharing) on the other. Mary Louise Pratt, however, emphasizes the tellability of scandalous events [956, pp. 136–147], though such material may actually reside at the precipice of actual taboo.

[8]Later, I propose an operationalization of this process by which a system may automatically identify tellable event sequences that emerge from simulations.

Moreover, as Ryan explains, and as others have as well [1297, 629, 959], many of storytelling's classic crosscultural tropes are rooted in such unrealized projections:

> Among the situations that involve virtual strings of events, we find themes of such common occurrence across cultures, periods, and genres as unsuccessful action, broken promises, violated interdictions, mistaken interpretation, and double, as well as single deception. [1066, p. 158]

Brilliantly, Ryan shows how these projections, diegetically embedded into the minds of characters, actually have a narrative structure (the poetics of which is itself a full-fledged narratology [1063, p. 328]). Ryan calls such cases *embedded narratives*: "story-like representation[s] produced in the mind of a character and reproduced in the mind of a reader" [1063, p. 320].[9]

David Elson argues that tellability is rooted ultimately in how a story modulates the basic needs of its characters through what he calls its *thematic content* [312, p. 45]. In Elson's *story intention graph*—by my estimation, the most deeply considered narrative schema in existence—character plans are represented as graphs whose sink nodes work as "affectively-charged termini" for those plans [312, p. 115]. By this configuration, story content is tellable when it *affects* a character, with regard to her basic needs.[10] This is particularly useful, since most

[9]In a very interesting ongoing project, Ben Kybartas and collaborators are developing a videogame that is driven by an operationalization of Marie-Laure Ryan's possible-worlds narratology. The game is called *Subject and Subjectivity* [626], and it is also showcases conversational interaction and text generation.

[10]In his dissertation, Elson identifies twelve such basic needs (motivated by the work of Abraham Maslow [752] and Manfred Max-Neef [790]): life, health, ego, wealth, love, leisure, membership, actualization, freedom, justice, enlightenment, and honor. Again, we find connection to Schank's categories of absolute interest, but also to Ryan's embedded narratives, inasmuch as character desires (and dreams and aspirations and fears) may pertain to such categories. It is worth noting, however, that content affecting such character needs will not automatically produce a good story, or a story at all. For instance, I believe that a fundamental problem with *Tale-Spin* is that its stories tend to be about uninteresting character goals, like a bird seeking a worm. While a bird's procurement of a worm will affect her health (recall Meehan's story criterion pertaining to bodily needs), that does not mean that a (good) story will necessarily obtain. To be clear, I do not mean to imply that Elson would claim as much—he would not—but it is worth noting nonetheless.

scholars of tellability aim to describe it as a sociolinguistic epiphenomenon of conversation, rather than to provide a framework for producing tellable stories.[11]

Finally, it is worth noting that tellability is ultimately established at the level of discourse, not in the reported event sequence. What this means is that good storytellers may render otherwise untellable events tellable, while bad storytellers may do the inversion.[12] Indeed, Jim Meehan has made this very point [824, p. 458]. Moreover, tellability is socially and culturally contextual, which means that a tellable event sequence may not always be tellable, even if the telling remains identical. For example, a riveting account of a strange dream will likely be better received by one's partner in the morning than in the middle of the night. When it comes to the prospect of automatic recognition of tellable events, the takeaway here is that the events cannot do all of the heavy lifting—ultimately, a horrible recounting of an interesting occurrence makes for a horrible story.

Simulating Toward Surface Intrigue

Let us return now to the specific concerns of simulation. While much of the knowledge derived from the scholarly work outlined in this section pertains to storytelling at the level of discourse,[13] there are still plenty of ideas that may help to ensure that a simulation domain has inherent intrigue. As such, these ideas should be of use to practitioners who seek to simulate storyworlds that may produce tellable event sequences in the first place: ones that, when recounted in a generated story, will not elicit the dreaded "So what?" that is famous in the

[11] Generally, the work on interestingness and tellability is descriptive, not prescriptive, but the high-level insights are still invaluable for simulation crafting toward compelling emergent narrative.

[12] In her PhD thesis, titled *Unputdownable*, Beth Cardier writes about tellability from the standpoint of a narrative practitioner [159].

[13] Accordingly, I will revisit these ideas later on in my discussion of the considerations surrounding curation in emergent narrative.

tellability literature [629, 949, 637, 1076, 634].

The most immediately pertinent of these ideas is Schank's notion of absolute interests, including topics such as danger, death, sex, money, destruction, chaos, power, and romance. Along such lines, perhaps the *clowns world* of Robert F. Simmons *could* produce narrative if, for example, the clown's actions could lead to him being in danger of death due to having sex for money. Likewise, we might imagine stories about *Elmer* the taxi driver's destructive scheme to plunge Simon City into chaos under the influence of his powerful lover. Of additional use are Schank's operators for personal relatedness and unexpectedness. While the notion of personal relatedness is an obvious ramification of the case of interactive emergent narrative—indeed, I discussed Schank's thinking in relation to interactivity in Section 3.1.2—unexpectedness may be utilized too. As an exercise, let us take the near story of Simmons and Slocum quoted above and minimally reformulate it according to Schank's ideas:

> Mary was wrestling with a bomb at the liquor bar before John helped her with it. John saw Mary wresting with a bomb at the liquor bar. John went over to help Mary with a bomb before he drew a wire. John and Mary together perished in the blast.

This modified example, I contend, is undoubtedly a story—I have rendered it *tellable*. Note that I only made three small changes: to serve the absolute interest of danger, 'bottle' and 'bottle cork' became 'bomb' and 'wire' respectively, and 'drank the champagne' was changed to 'perished in the blast' to target the unexpectedness operator taking the absolute interest of death as its argument.[14] Now the latter move is one of curation—it pertains to the telling of the story—but

[14]Jonathan Lessard, who is on the reading committee for this dissertation, was not impressed by this modified tale: "Not convinced your updated story is better. It's just more conventional. I adhere more to your earlier argument on 'art brut' and the weirdness of the naïve stories of systems" (personal communication, July 27, 2018). While here I am attempting to make a certain point, one that I will reveal momentarily, I think I probably agree with Jonathan. Sometimes I do not know how to feel.

unexpectedness at that level must be undergirded by unexpectedness at the level of simulation. If a simulation of Mary and John's attempted detonation at the bar does not include the real prospect of them failing and dying as a result, then unexpectedness at the level of surface expression is probably already a lost cause.

Connecting back to Behrooz's common themes and *Brutus*'s organizing principle, it is possible to target tried-and-true literary themes like underdog stories, narratives of betrayal, and so forth. Again, this is a matter of sculpting the simulation's narrative possibility space—particularly as it is yielded by character actions—to support emergent phenomena that matches these themes. As I will express in Part II of this dissertation, this seemingly obvious notion mostly escaped me for the first several years of my simulation practice. While early on I did model (either explicitly or implicitly) some basic Schankian absolute interests (sex, danger, death), it is at a higher level of dynamic interaction between the actions that produce such intrigue that literary themes such as betrayal emerge. As I note in my discussion of the *Hennepin* simulation engine in Chapter 11, my first authoring goal with that system was to sculpt the possibility space such that *revenge* stories could emerge (and in particular an evocative set of possibilities with arson as *modus operandi*).

Lastly, a simulation may also be crafted to target Ryan's examples of situational tropes rooted in embedded narratives: unsuccessful action, broken promises, violated interdictions, deception, and so forth. Mads Wilbroe and collaborators use Ryan's ideas to discuss a system of theirs [645] that could not generate tellable events because no alternative possible worlds could be suggested [1336]. Indeed, this seems to require some modeling of the internal worlds of characters—their beliefs, their desires, and to also connect back to Elson, the interplay between action and character wellbeing. In Part II of this dissertation, I will discuss my own methodology for modeling these phenomena inasmuch as it has evolved across the development of each of my three simulation engines.

As noted above, several researchers have suggested that unique event sequences are often tellable. Thus, if a simulation yields a large possibility space in terms of emergent event sequences, it might be that the less frequently occurring sequences will be tellable with regard to the simulation domain. Of course, this may not be the case, since the uniqueness of a sequence is in the eye of the beholder: unusual events in the real world are only tellable if the recipient also assigns a low probability of occurrence to those events.[15] In the case of a system telling stories about a simulation, this very pattern could occur if the system omnisciently identified an objectively unusual sequence and then told a story recounting it to a human recipient with no refined sense of what is usual or unusual in the simulation's possibility space. While this is a pitfall, if a recipient could in fact develop a probabilistic model for that possibility space, this would be a huge win, since it is trivial to identify unusual sequences.

Indeed, Morteza Behrooz and collaborators have proposed this very method and explored it in the context of gameplay logs from a rummy videogame with chat interaction [96]. Now Behrooz is exploring how domain features may be used to build semantically richer models of domain interestingness that may be used to identify tellable event sequences [96].[16] In the case of crafting a simulation, an author can consciously determine what would be interesting in that domain to

[15]Such probability mismatches are likely at the root of tellability errors in child storytelling: due to inexperience, the child has not yet developed a robust probability model for events, and so events that seem unusual to them (and thus tellable) may seem usual (and thus not tellable) to adult recipients with refined probability models. I think a Bayesian study of tellability could prove fruitful, and indeed such methods have recently emerged in literary analysis [622].

[16]As I explain later, in curationist terms, Behrooz's work that I have just referenced concerns the development of *sifting heuristics*. These are heuristics that a *story sifter* module may use to automatically identify interesting event sequences that emerged in the course of a simulation. When such a sequence is identified, the recorded material associated with it is handed off to a *narrativizer* module that constructs a narrative artifact recounting the sequence.

both support the emergence of interesting material and then omnisciently detect such material as it does emerge. This kind of omniscience (and omnipotence) is a core appeal of building one's own task domain, but typically that comes with the drawbacks of the *toy domain*—these drawbacks are what lead to *interactionist AI* [658]. In the case of story generation, however, the drawbacks of toy models are avoided: as Rodney Brooks famously said, "the world is its own best model" [144, p. 5], and in story generation we might say that *the model is its own best world.*[17] That is, it is not a proxy for the real task domain, or at least that is not a common construal.[18]

Toward Personalized Narrative

Finally, though in this case I do not practice what I preach, I think it is worthwhile to discuss the prospect of crafting a simulation that may support Behrooz's notions of topic interest, reminiscence, and implicit familiarity. These dimensions critically rely on subjective characteristics of the listener: who she is, what she likes, what her life has been like, what signs hold symbolic weight for her, and so forth. A simulation whose generated stories could engage a person in such highly personal ways would be an example of *personalized narrative.* In personalized narrative, story generation proceeds from a model of an individual for whom a narrative artifact is specially generated such that the experience of

[17] I owe this rephrasing to Ian Horswill—a former Rodney Brooks student [512], and a member of the reading committee for this thesis—who used it to characterize my argument (in Section 3.1.1) that simulated worlds are in a sense real.

[18] Sheldon Klein realized this in his pioneering early work on story generation. In his first note on "automatic novel writing", he mentions the appeal of working with a "universe in which the non-linguistic data are completely known and controlled" [590, p. 417]. We might consider, however, what it would mean for a story generator to have our real world as its storyworld—the result would be something akin to *procedural nonfiction*: procedurally generated stories about real-world phenomena. This relates to the concept of *narrativization*, the generation of narrative accounts of raw data, which I discuss periodically throughout this document. Examples of procedural nonfiction include *Terminal Time* [768, 780, 268] and *StatsMonkey* [29].

engaging with it may be a reflective one that allows her to mediate on herself and on the world [782]. My colleague Melanie Dickinson is exploring this area as her primary research agenda here at the Expressive Intelligence Studio. For her, the personalized mode of story generation has the potential not only to produce highly compelling experiences, but to also serve as a facilitator for positive change at the level of individuals, organizations, societies, and humanity [263].

Personalized narrative is still a very young area, however. One line of work has explored the integration of *player modeling* [1168, 1361] into interactive storytelling systems. This started with the development of experience managers that consider the history of player actions [398, 1326, 1237, 776, 724], but beginning with David Thue's *PaSSAGE* (2008) [1252, 1251], systems have sought to also model player preferences [568, 976, 1253, 477, 580].[19] One particularly relevant example of personalized narrative is BBC R&D's ongoing project *Visual Perceptive Media* [413, 321, 218], the outputs of which include *Breaking Out* (2012) [349, 189], an audio drama whose content varies depending on time and geographic location of the listening session, and the short film *The Break-Up* (2015) [360], which dynamically adapts its music, video, and color grade to the viewer. At the 2017 International Conference on Interactive Digital Storytelling, keynote speaker Pia Tikka discussed ongoing work in the area of *neurocinematics* [459, 1256, 1255] that investigates the prospect of cinema that dynamically

[19]Though lesser known, an earlier paper by Piotr Gmytrasiewicz and Christine Lisetti (2000) proposed an architecture that would model the player's emotional state and alter gameplay in response [396]. An intriguing alternative to this kind of personalization is the "inverse user model" [759, p. 2] of Michael Mateas's *Subjective Avatars* project of the late 1990s [757, 759, 760, 762]. Instead of learning a model of the player in an effort to fit the story to her preferences and sensibilities, the system subjectively models the player's avatar to have its own internal life and then "actively manipulates [the] user's experience so as to try and make the user feel the same way as a character" [759, p. 2]. Earlier, in the first tech report on the Oz Project (the tradition in which Mateas was operating as a graduate student), Sean Smith and Joe Bates asked the reader to "Consider worlds where the user is the homunculus inside some other character's head—the user interacts through this different personality and partially assumes it" [1179, p. 2].

adapts according to a viewer's brain activity [1257, 884]. One practitioner and scholar in this nascent area is Richard Ramchurn, whose collaborative work on the subject includes several papers [944, 974, 172] and the "brain-controlled movies" *#Scanners* (2015) [943, 942, 945] and *The Moment* (2018) [975]. For general discussion on the prospect of personalized media, see these three recent publications [782, 1078, 502].

As a future direction, I think practitioners could work to evoke in a particular individual Behrooz's sensations of topic interest (model or target her favorite topics), reminiscence (connect to her past experiences), and implicit familiarity (develop a sign system by which semiotic entities that are cognitively rich for her are mapped onto modeled entities in the simulation domain). The latter effect appears to be the most challenging one to produce, but we have a head start. In the *quantified self* movement [718], for example, we find an interest taken by individuals to model themselves in a way that could be conducive to the kind of personalized narrative that is envisioned by Dickinson and suggested by Behrooz's notion of implicit familiarity. Furthermore, in the work of thinker and practitioner D. Fox Harrell, we find the application of *algebraic semiotics* to computational narrative [400, 452, 453], which is precisely a method for operationalizing the kind of phenomenon characterized by Behrooz's example of a red rose that evokes a visceral sense of familiarity. I can imagine a project like Dickinson's that uses an analogic representational scheme of the kind used in Harrell's work to map between the kind of data captured in the quantified-self movement and the particular entities modeled in a given simulation domain.

4.1.2 Granularity Extremes

The *granularity* of a simulation pertains to the level of detail of the modeling of simulation entities and processes. In the case of emergent narrative, critical dimensions of granularity include the modeling of space and time, the representation of characters, the breadth of possible character actions, and the level of detail with which a typical character action is modeled. As I will show in this section, works of emergent narrative may be troubled by simulations whose granularities are either too low or too high.[20]

Life and Low Granularity

Let us begin with some brief discussion of the more intuitive problem: a simulation with too little detail. In the extreme case, it may not be possible to produce anything that would even be called narrative. An example of this might be something abstract like John Conway's *Game of Life* [207], though the scholar Richard Walsh has argued that the terminology of 'guns' and 'gliders'—famous emergent structures in the world of *Life* hacking [10]—actually constitutes a kind of narration [1308, p. 75]. Nonetheless, by any interpretation, it is probably unreasonable to say that Conway's *Game of Life* produces *good* stories. Later, in Chapter 7, I

[20]In an earlier draft of this thesis, I used the term 'fidelity' in place of 'granularity', since I have heard the former used more often to refer to this concept. However, Noah Wardrip-Fruin brought up a good point in his feedback on that draft: "I'm not sure the word 'fidelity' is what you want in [this section]. The word, for me, is strongly associated with closeness to an existing phenomenon. But you are also interested in the extent to which novel phenomena (unique to the generative world) are modeled, right?" (personal communication, July 3, 2018). This issue pertains to the curious usage of 'simulation' to refer to systems that do not model subject phenomena. As I discussed in an earlier footnote, in his paper "Do Videogames Simulate? Virtuality and Imitation in the Philosophy of Simulation" [567], Veli-Matti Karhulahti has decried this usage—which seems to be germane to videogames—because it violates the established scientific definition of the term, which necessitates that a system imitate some real one. By this classical definition, it is natural to discuss the fidelity of a simulation, but this is not the case with the colloquial sense, which I use in this document. This is why I am now using 'granularity' instead of 'fidelity'.

will critique my first simulation engine, *World*, for its low modeling granularity.[21]

***Saga II*'s Drinking Machine**

Now, let us consider the less intuitive case: simulations with too much detail. Here, it will be illustrative to consider the example of what it currently to my knowledge the earliest work of emergent narrative: *Saga II*.[22] In 1960, CBS produced a television miniseries, called *Tomorrow*, that was intended to honor the upcoming centennial of MIT. One episode in this series, titled "The Thinking Machine", focused on the cutting edge of work being done with computers. As the concept for the episode was being developed, a CBS producer named Thomas

[21] From a different perspective, cellular automata might be deemed extremely *high*-granularity, because the modeled components are so small and modular. This claim makes sense when considered in the context of *artificial life* [641] projects that seek to generate lifelike structures or behaviors that emerge from the low-level patterns of cellular automata [640, 185, 1121]. This is a remarkable challenge, however, and the resulting "digital creatures" are typically insectoid [319]. Some practitioners in artificial life do attempt to generate narrative, but by eschewing cellular automata to instead start from a higher-order agent representation. A remarkable undertaking in this area is the *autobiographic agents* project of Kerstin Dautenhahn, Chrystopher Nehaniv, Wan Ching Ho, and others [240, 874, 875]. Autobiographic agents are autonomous agents who dynamically construct life histories composed of symbolic "memories" of their past experiences; these autobiographies crucially guide (and may be revised according to) activity carried out by the agents as the simulation proceeds [873]. In this project, autobiographic memory is broadly considered as an adaptive survival mechanism, an idea that may tested through simulations that explore the extent to which agents with autobiographic facilities survive longer than purely reactive agents in simulations with resource competition [489, 490, 488, 491, 260]. Intriguingly, these researchers have also experimented with agents who may *communicate* their autobiographical memories to other agents [493, 492]. My original dissertation proposal [1038] was likewise about autonomous agents who narrativize their ongoing experiences in a simulated world, and in general that notion corresponds to a particular *subjective* mode of curationist emergent narrative. This connects nicely to a fabulous paper by Jonathan Lessard and Dominic Arsenault, titled "The Subjective Interface" [681], that proposes a design pattern whereby the ground truth of a simulated storyworld may only be accessed through the subjective interface of a potentially unreliable character.

[22] In fact, to my knowledge this is more broadly the earliest story generator of any kind [1039]. Moreover, I believe it to be the earliest major project in computational art, since the depth and complexity of its computation (and the number of person hours expended in its development) greatly exceed that of earlier projects like Christopher Strachey's love-letter generator [1205, 1312] and Theo Lutz's poetry generator [719, 364]. I first learned of the system in a paper by Mark Sample [1082]. With Gillian Smith, I am currently undertaking an excavation of *Saga II* that has already turned up an abundance of archived materials. (We hope to report on our dig soon.) Finally, Liza Daly has recently produced a playful reimplementation of *Saga II*, which she named *Saga III* [236].

Wolfe suggested that the TX-0 computer at MIT could be used to generate a script for a TV western [1349, 1183]. In response to this challenge, MIT researchers Douglas T. Ross and Harrison "Dit" Morse took on the challenge and spent several months developing a system called *Saga II* [859].[23] Rather than entire television episodes, the system generates choreography for a shootout pantomime between a sheriff and a robber at a hideout. In the actual CBS television show, which was viewed in an estimated 7.7M homes [1028], three of the system's outputs were showcased as professionally produced playlets. (I explain more about this aspect of the project in my discussion of curation in Chapter 5.)

It is interesting to consider the design decisions made during the development of *Saga II*, because they represent what appears to be a first foray into emergent narrative. Ross and Morse had no earlier works to consult in deciding how their project would work, so they were forced to invent an approach to character simulation out of whole cloth.[24] In building such a simulation, a primary consideration is granularity, and Ross and Morse chose to represent the modeled action in extreme detail. Specifically, *Saga II* is very detailed in its simulation of characters handling objects: the system keeps track of what is in each of the character's hands, and picking up an object may entail a complex procedure by which objects that are currently held by a character are transferred between a character's hands and or the modeled environment. For example, there is a bottle of whisky in

[23]One of the open questions in our excavation project is whether there was an antecedent system to this one—that is, why the 'II'? Doug Ross, who coined the term *computer-aided design* [1030] and led development of the Automatic Programmed Tools language [1029], passed away in 2007. I am currently attempting to make contact with Dit Morse, as I did in a recent project in which I contacted Joe Grimes as part of an excavation of his forgotten 1960s story generator [1039]. A letter to his last recorded address was returned.

[24]Even if such an antecedent exists—and please let me know if one does—it is clear that the *Saga II* creators would not have been aware of it. Generally, I think it is more fruitful to consider cases of independent reinvention as though they were cases of first invention—that way, each may provide a different view of the broader phenomenon of that invention. This is the approach I am taking in a history of interactive branching narrative that is currently in development.

the hideout, which the characters can pick up, move, pour from, and drink from. These affordances drive one of the more interesting features of the system, which its creators call the "inebriation factor":

> Every time the sheriff or the robber takes a drink—and we programmed opportunities for doing this—the inebriation factor is increased, making intelligent actions a little less probable and unintelligent actions a little more probable. After a few drinks, a man takes longer to do things. After the computer's robber has had several drinks, instead of aiming only once to fire, he often aims three times before firing. [963, p. 3]

As Figure 4.2 illustrates, the procedure for having a drink can be quite complex, entailing a number of steps along the way.[25] For instance, if a character is holding a gun in his right hand in the corner of the hideout, to take a drink out of a glass he will have to follow this convoluted procedure: move to the location of the bottle and the glass, transfer the gun to the left hand, pick up the glass with right hand, transfer the gun from left hand to a surface at current location, transfer the glass to left hand, pick up the bottle with right hand, pour whisky from the bottle in right hand into the glass in left hand, transfer the bottle from right hand to the surface at current location, transfer the glass from left hand to right hand, at which point a drink may finally be taken from the glass.[26]

[25]The numbered circles in this diagram represent probabilistic switches that route the simulation processes: when such a switch is encountered, a probabilistic distribution over its branches is computed according to the state of the storyworld, and then a pseudorandom number is generated to determine which branch to probabilistically take. Intriguingly, an interactive mode allows a user to adjust, at runtime, the procedures that determine the various branching probabilities. This is the subject of the creators' only report on the *Saga II* project, an internal technical memorandum written by Morse [859]. This mode was developed not for use by interactors, but as a way for the authors to quickly tune system parameters in the midst of impending CBS-TV production deadlines. Thus, one might view this interactive mode as the first authoring tool in procedural narrative. Currently, Yotam Shibolet and collaborators are carrying out a project to ascertain the history of authoring tools for interactive storytelling [1144]; I look forward to reading about their findings.

[26]Note that a character may also drink straight from the bottle, which will cause his inebriation factor to increase by a double increment. Intriguingly, the more a character drinks, the more he wants to continue drinking, and so in some generated stories the robber falls into a drunken

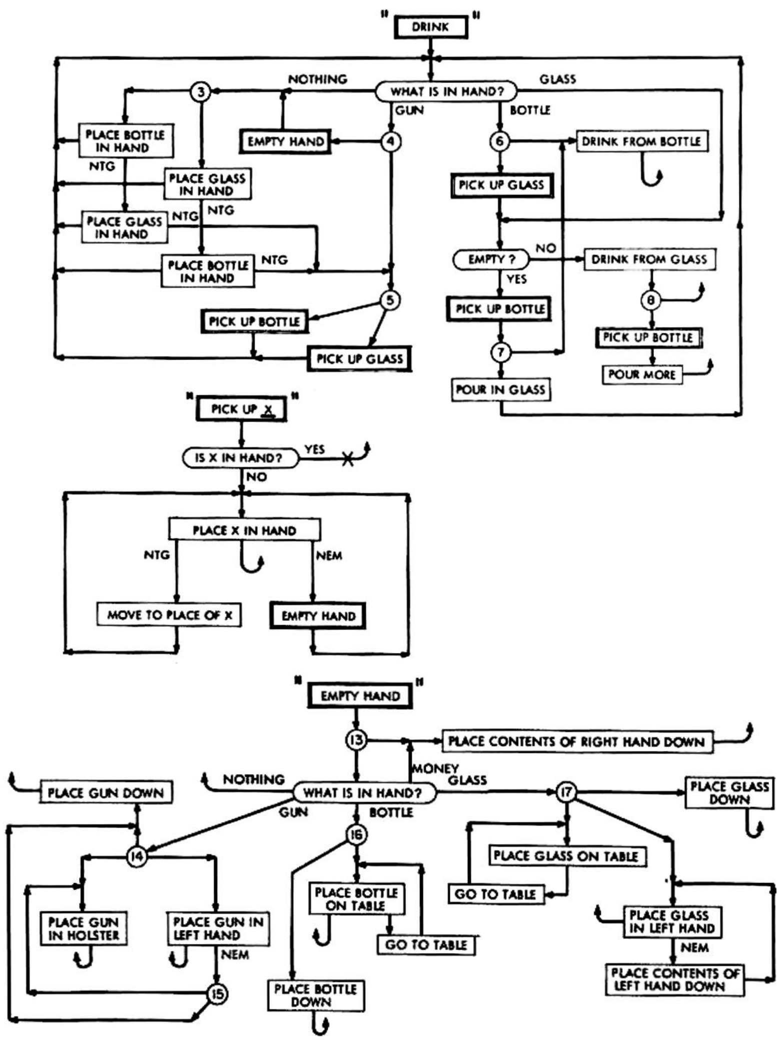


Figure 4.2: *Saga II*'s drinking machine. Perhaps the earliest effort in emergent narrative, *Saga II* (1960) is a system that generates stories in the domain of the TV western by simulating a shootout between a sheriff and a robber. One interesting component of the project is the modeling of an 'inebriation factor' by which characters who drink from a whiskey bottle may act less rationally. This excerpt from an original flow chart illustrating *Saga II*'s processes shows how the act of drinking works in the simulation. Because the system models the handling of objects in extreme detail, these processes compose a rather complex code machine that would have taken considerable authoring effort to produce (for arguably incommensurate payoff). This example illustrates how simulation granularity in emergent narrative can sometimes be too high. (Courtesy Computer History Museum)

stupor before the sheriff even arrives at the hideout. In one experiment, the creators turned up the urge to drink, and instead of having a gunfight, the sheriff and the robber spent the scene vying for the bottle. Another humorous output ended with the robber killing the sheriff and having a victory drink, and then eight more, before leaving the hideout without his gun [962].

While *Saga II*'s detailed simulation of prop handling can lead to some mildly interesting emergent phenomena, the considerable authoring effort entailed was probably not merited. Moreover, this extreme level of detail would likely be unsustainable for a larger-scale storyworld. As Figure 4.2 illustrates, the simple case of a character taking a drink required the authoring of a fairly complex code routine; the other phenomena modeled by the system are driven by procedures authored in this same way. That being said, one could argue that the system is specifically a machine for generating gunfights, and so the level of detail actually nicely fits the domain. That is, gunfights are all about the speed of the draw, and this depends on the ability of a character to quickly handle a gun, and such behavior depends primarily on the handling of objects. Indeed, this rich modeling does lead to some interesting emergent phenomena in *Saga II*: for example, a character could be shot because he had placed his gun in his holder to take a drink. While this is perhaps the case, my sense is that the authoring effort that was exerted to model the details of object handling could have instead yielded emergent phenomena with more narrative intrigue. For example, the inebriation factor itself required little authoring effort—drinks increment a number that alters the probability of taking certain subsequent actions—and produces far more interesting emergent situations than the object handling. While this depends on drinking, which in the implementation depends on object handling, Ross and Morse could have instead modeled drinking at a higher level of abstraction (e.g., a character must just be near the bottle to drink). If Ross and Morse had instead worked primarily at that level of abstraction, they may have yielded an even richer possibility space. That being said, *Saga II* is a remarkable first foray into emergent narrative, and in my estimation an extremely successful project. Later, I will discuss how impressed I am by its level of curation.

High Granularity in ***Tale-Spin***, ***Prom Week***, and ***Dwarf Fortress***

It is not hard to name projects in emergent narrative that have attempted to wrangle a simulation whose granularity is probably too high. In a sense, the hallmark of emergent narrative is modeling a storyworld with remarkable detail. Other examples include *Tale-Spin*, whose character actions correspond to the primitives of Roger Schank's *conceptual dependency theory* [1101], which means its simulated events may for instance pertain to `propel` and `ingest` transformations on objects. At this level of granularity, it takes a lot of authoring to ensure that interesting stories may actually emerge at a higher order than the physical primitives.

In a related example, *Prom Week* [799, 803] cocreator Mike Treanor has come to believe that the game's underlying social simulation was probably too complex. I asked him about this over email, and Treanor noted that a simpler social schema [804] would have dramatically reduced the project's considerable authorial burden:

> So we had about 5000 "social considerations". [...] I think the biggest reason we ended up with so many rules is that we weren't as careful about what we put in our social schema [...] We broke things down into 3 binary reciprocal relationships, 3 undirected weighted relationships (networks), maybe like 15 statuses, like 25 traits, 10 or so history labels, and 10 or so cultural knowledge and labels about possession of cultural knowledge. Each one of those features needed to be considered with regard to all combinations of each of the other features [...] So, how could we have gotten away with fewer rules? The easy answer is that I think we could have had fewer traits, fewer statuses, etc. Because of exponential relationships these things have to one another, each one you remove reduces authoring considerably.[27]

Of course, a reduction in simulation complexity would likely come at the cost of emergent intrigue, but Treanor believes that the extreme detail of the game's

[27] Personal communication, April 23, 2018.

social simulation was overkill with regard to the amount of complexity that players would actually come to comprehend through gameplay:

> But also, once the game was finally finished (with the current interface with the thought bubbles), we actually ended up creating intermediate abstractions to help reduce the amount of things that the player would need to think about. A goal, of mine at least, in Prom Week was for the player to actually think about and care about what [the underlying AI system] *CiF* was doing, and in the end, we needed to reduce it (the contents of the thought bubbles are a synthesis of [social] rules). So, through playtesting, we could have probably figured out what the level of abstraction that the players would be able to actually comprehend was, and had that be what the [social] rules were about (rather than our initial, overly robust decomposition of social reality).[28]

Of course, there are exceptions to every rule, and here *Dwarf Fortress* is a strong one, since its success has largely been predicated on the extreme granularity of its simulation (a hallmark that is often cited in its various glowing reviews [1321, 1344, 442]). However, its creators' solution to immense authorial burden has been to carry out a thirty-year project [1027], and cocreator Tarn Adams has himself warned of the danger of excessive detail in terms of its exacerbation of authorial burden (and confusion):

> There's no reason to have 50 variables modeling one aspect of a game's behavior if they don't all have a meaningful impact on the game. Operate at the level of what the player sees or one layer below. Don't get carried away and create a lot of worthless noise. Not only do uselessly complicated systems hinder tuning, but they can have a paralyzing effect during development. Use complexity only where it's needed, otherwise strive for simplicity. [15, p. 520]

In my own work, I have struggled at various points with both granularity extremes, as my self-critiques in Part II will show. Granularity poses a tough design challenge, since homing in on the right level is a process of discovery that

[28] Personal communication, April 23, 2018.

entails a lot of iteration and tuning. In his email, Mike Treanor remarked as much following his explanation of *Prom Week*'s level of detail:

> But there was no way we could have known any of that at the time! [...] I don't know, we just gotta make a lot more of these things in order to make any sense of authoring for it.[29]

4.1.3 Low Modularity

Some character simulations are not modular enough. Though this pitfall may also manifest in adverse surface effects, as I will explain, it relates primarily to the authorial considerations of simulation crafting. Here, I am specifically concerned with the modularity of the procedural content that defines how the storyworld is represented (for instance, how characters are modeled) and how it may evolve (namely, character actions). Modularity in this sense increases as monolithic chunks of procedural content are broken into smaller pieces that may be recombined in a variety of ways.

Klein's Misunderstood System

Modularity is low, then, when the simulation's procedural content is structured as a collection of relatively monolithic chunks. As an example of this, we might consider the murder-mystery generator developed by Sheldon Klein and his students in the early 1970s, the modularity issues of which have been cited as the impetus for the tidal shift in story generation that lead to the predominance of planning techniques.[30] As Noah Wardrip-Fruin explains, "the field has generally followed Meehan's critique of the Klein's system: it was made up of explicit chunks

[29] Personal communication, April 23, 2018.

[30] Klein's system has for decades been known as the earliest effort in story generation, but in a recent paper [1039] I reported on an excavation of three forgotten earlier efforts: the aforementioned *Saga II* of 1960 [859], Joseph E. Grimes's unnamed folktale generator of circa 1963 [2, 423, 422, 424], and Robert I. Binnick's unnamed folktale generator of 1969 [110, p. 27].

of action, with a path through these chunks selected randomly" [1311, p. 116].[31] Indeed, it is not hard to find recapitulations of this criticism by contemporary scholars [693, 540]. Though there is some veracity to it, the origin of the critique seems to be rooted partially in misunderstanding.

While many have characterized Klein's system as a story grammar or template-based system [1363, 1366, 1195, 329], that is very far from the case, as I will now show.[32] More aptly, the architecture should be described as a *production system*.[33] In a production system, the fundamental unit of procedural content is the *production rule* (or just *rule*), which is a package comprising two components: a

[31]Note that there are actually two components to this critique. The first relates to the modularity of the procedural content, referred to by 'chunks of action,' and the second to a lack of abstraction in designing that content, denoted by 'explicit'. In this section, I discuss modularity, and in the next I discuss the related issue of abstraction.

[32]In a *story grammar*, stories are generated by terminally rewriting the start symbol of a context-free grammar whose nodes correspond to either plot events or segments of a natural-language narrative telling [1034]. As you will read momentarily, this is a ridiculous characterization of Klein's system. I have a few theories for how it may have emerged, though. First, Klein illustrates the workings of his system in figures that use *Backus–Naur form* [601, pp. 6–8], which is a grammar-like notation that is commonly used to define the operation of programming languages [808]. To the uninitiated, this could be seen as an indication that Klein's system has the power of a context-free grammar, but it is in fact far more powerful. Second, Klein was a participant in the *text grammar* movement of the 1970s [592], which sought to develop structural accounts of meaning at the level of discourse (thus, 'text' as a discursive object, not a string of symbols), as an opposition to the Chomskian fetishization of the sentence [1284]. This movement had its locus in Europe, and I believe unfamiliar American scholars may have assumed it was concerned with story grammars. Lastly, story grammars were the subject of harsh critique by scholars whose associates sought to distance themselves from Klein's work: story grammar's co-executioner Robert Wilensky [116] was a labmate of Natalie Dehn and Michael Lebowitz [252, 665], who led the field of story generation in the early 1980s and who, in seeking to substantiate their own contributions, may have found it convenient to reductively conceptualize Klein's system as a kind of story grammar. Lastly, I will note that while the misconception is predominant, not all scholars have succumb to it [1065].

[33]Roger Midmore has argued that Klein actually invented the *blackboard architecture* [833, p. 9], which is related to the notion of a production system, though Klein's inception was in an ancestor to this system [585, 588]. Production systems are believed to have been first described by the mathematician Emil Post in 1943 [954], building off his earlier work [953], though Pāṇini's ancient generative grammar (with its rules about rules and other mechanism) is a notable antecedent [1191, 571, 108]. In any event, computerized production systems were just emerging [245] as Klein began to incorporate them into his own technical practice [585, 588, 589]. He does not use the standard terminology of production systems, which suggests that Klein may have independently invented the concept. For the sake of clear explanation, I will use the conventional terminology here.

list of *preconditions* specifying aspects of the system state that must hold in order for a rule to be executed, and a list of *effects* (also called *postconditions*) that define how the system state will be altered if the rule is ultimately executed. In a production system, the portion of system state that may be altered by production rules is represented as a knowledgebase containing *facts* about the world, which is called the *working memory*. Typically, a rule is tested by evaluating each of its preconditions—if all of them hold, the rule is *triggered*, and subsequently all of its effects will be *executed* to update the working memory. Klein, however, introduces an additional wrinkle. In his system, authors associate preconditions with *weights* (probability increments or decrements), and the result of testing a rule is a probability of the rule actually being triggered. That is, as each precondition is evaluated, the sum probability of triggering the rule is either incremented or decremented according to the weights associated with the preconditions.[34] Once a sum probability has been computed, a pseudorandom number (between $0.0 - 1.0$) is generated, and if it is less than the computed probability (treated as a floating-point number), the rule is triggered, and thereby all its effects are executed. By executing effects, new facts may be added to the working memory and existing facts may be removed.[35]

[34]Note that preconditions in Klein's system may bundle multiple predicates, with variable binding across the predicates. For example, the rule `GEORGE KNOWS &` $\neq$ `HATES &` $\neq$ `DISLIKES Y` [601, p. 14]—which may be read as 'George knows and does not hate and does not dislike Y'—is a composite of the three predicates $knows(George, Y)$, $\neg hates(George, Y)$, $\neg dislikes(George, Y)$, with Y bound consistently to the same entity across the predicates. An *or* operator may similarly be used. Additionally, each precondition is more precisely associated with two probability increments or decrements: one that is used when the precondition holds and another that is used when it does not. Finally, effect execution can be made deterministic, rather than probabilistic, by simply using numbers outside the range $0.0 - 1.0$. That is, if the weight 10 is associated with a precondition, and that precondition holds, then the effects in the rule will certainly be executed (unless, of course the author also associates huge negative weights with other conditions in the package).

[35]To be a precise, in lieu of removing a fact, a new fact representing its negation is added to the working memory. Additionally, the working memory keeps track of how long it has been since each fact was asserted, which allows for preconditions to check how much story time has passed since an action occurred or a stative was asserted (or negated). Because the deletion of

By hacking his production system to work probabilistically, Klein's architecture cleverly approximates the expressive power of *utility-based action selection* [723, 742], while still keeping within the computational limitations of the day. To illustrate this, let us compare it to *Comme il Faut* (or *CiF*, pronounced 'siff') [804], the utility-based AI architecture that underpins the videogame *Prom Week* [799, 803]. While in Klein's system precondition evaluation derives a probability of executing a list of effects, in *CiF* precondition evaluation derives a utility of taking a prospective action. (For our purposes, executing a list of effects and taking an action may be treated as being equivalent.) In *CiF*, the utility of all candidate actions is computed before an action is taken (the top-scoring one), which provides guarantees that the most fitting action (as implied by the authored rules) is always taken. This guarantee necessitates the evaluation of potentially thousands of preconditions, which in Klein's day would not have been feasible. In one paper, Klein warns of the danger of evaluating even a few dozen preconditions:

> Then, for each X and Y pair, evaluate the [preconditions] for this rule. [...] Note that if the class specified for the X variable contains 8 objects, and that for the Y variables contains 5 objects, the [preconditions] for this rule will be evaluated 40 times. Such nested loops can be very time consuming and should be used with caution. [601, pp. 13–14]

Thus, the naïve solution, which is moreover a common operating mode for production systems, is to trigger the first rule whose preconditions all hold. But in return for its speed gains, this method sacrifices any hope of the optimal action being taken, since the corresponding production rule may not be the first one whose preconditions are evaluated. Klein, however, designed an intermediate approach that uses author-defined probabilities to ensure that more fitting actions

a fact is represented by the addition of a new fact that negates the original one, the system can also reason about the duration for which statives were maintained, including the specific times (diegetic timesteps) of onset and termination. This affords both powerful authorial control (at authoring time) and expressive system reasoning (at runtime).

(given the current state of the storyworld), though not guaranteed, will be more *probable*: instead of deterministically triggering a rule whose preconditions hold, Klein's execution probability allows his system to potentially bypass the rule to choose another one whose preconditions also hold. This provides an approximation of utility without necessitating the evaluation of all rules before deciding which one should be triggered. As explained in Section 11.2.7, I utilize this same technique in my current simulation engine, *Hennepin*, since its expansive number of characters makes it susceptible to the computational limitations of our current day.[36]

Additionally, in Klein's architecture rule effects may alter the operation of the production system itself. Rules are associated with *frequencies*, where rules with higher frequencies are tested more often as story time progresses, and there is also a *priority* policy that determines the order in which rules will be tested. Intriguingly, the effects of executed rules may *disable* (or re-enable) other rules—disabled ones will not be tested—or modify their frequencies and priorities [597, p. 15].[37] This dynamic reformulation of the operation of the production system according to the effects of its own rules is a kind of *computational reflection*, which is a design pattern that characterizes computer programs that can read and modify their own code [1169]. Later examples of the use of this technique in computational narrative include extensions to the Oz Project's Hap language [992, 1127], *Façade*'s ABL language (whose *meta-behaviors* use reflection similarly

[36]The earliest writing on utility-based action selection in story generation that I am aware of appeared on the `rec.arts.int-fiction` newsgroup [342] in 1991, where it was proposed by Robert Taylor Fisher, an undergraduate at the California Institute of Technology who was working on interactive storytelling at the time [341]. An early implemented example by Brad Rhodes and Pattie Maes, based on Maes's *behavior network* architecture [723], appeared later in 1995 [997].

[37]Additionally, an author may organize rules into groups, which allows for the modification of activation statuses, frequencies, and priorities at the level of groups. Rule effects may also dynamically reconstruct the organization of rules into groups by forming new groups, removing groups, and changing the group memberships of individual rules. Again, this affords immense authorial power, in this case to a level that surpasses that of nearly all systems, historical or contemporary, that I am aware of.

to how it is used in Klein's architecture [764, pp. 93–96][773, p. 41]), and the Gertie language developed at Zoesis in the 2000s [708, p. 62].[38]

Clearly, Klein's system is no story grammar. So, how does it generate stories? In this architecture, the working memory is the very representation of the storyworld and its accumulated history.[39] In a beautiful expression of how nascent the concepts of story generation and production systems were at the time, Klein calls such storyworlds "stochastically modifiable semantic models of arbitrary universes", and the stories that are generated thereby "reports on the status of the modelled universe" [601, p. 1]. To generate such a report, the system uses special rules that match against collections of facts in the knowledgebase to target other rules in a context-sensitive grammar, whose execution produces strings that render the storyworld facts in natural language. Because facts in the knowledgebase are represented as triples of the form `(subject, relation, object)` and

[38]As Michael Mateas notes in his dissertation, "powerful features such as reflection, which edge towards 'you can do arbitrary things', are paradoxically useless without idioms providing guidance on how to use the feature" [764, p. 94]. One such idiom in Klein's framework is the deactivation of rule groups that are associated with beats in the story that have already passed. For example, after the murder occurs, the rules for playing tennis or going to a pub may be deactivated. This both saves on computation time, since those rules will not be needlessly evaluated after becoming obsolete, but it also provides an *authorial affordance* [764, pp. 125–126] that enables an author to conceptualize higher-order patterns that group character behaviors according to notions that might be called scenes or *beats* [818] or *social practices* [323, 326, 1268, 327]. In later work, Klein further explored reflection by introducing effects that not only reformulate the operation of the production system, but *generate* entirely new rules. In turn, these generated rules could contain effects that generate more new rules, and so on. Klein called this concept *meta-compiling* [592]. It is worth noting that Klein's system may not have been the first in computational narrative to use reflection: due to its vintage, *Saga II* likely used the method merely due to being programmed in assembly for a computer with limited memory. In later examples, however, reflection obtains also at the level of the *rhetorical machine* of an AI system [765], where it becomes a human-understandable computational move that may underpin specific authorial affordances, such as Klein's rule-group control. Design patterns like reflection feature heavily in the aesthetic space of *weird languages* [771].

[39]Later story generators that are also driven by production systems include *Telltale* [1157, 214] (by Alfred Correira and Robert F. Simmons, who was Klein's 1960s collaborator [1158, 604, 603, 1159]), which also seems to be misunderstood as a story grammar [1170, p. 303][1339, pp. 8, 18][1128, p. 3], and an obscure 1986 system by Marie-Laure Ryan [1064]. Additionally, the aforementioned *Prom Week* [804] utilizes production systems [770], as does *The Sims 3* [324] and *Versu* [326], for each of which Richard Evans served as an AI lead.

tuples of the form `(subject, attribute)`, the working memory yields a *semantic network* that represents entities (e.g., characters, locations, props) as *nodes* and associates them with their attributes and also connects them to one another according to relations (e.g., *loves*, *hates*) that are represented by *edges* in the graph.[40] Additionally, authors may specify composite nodes in the network that group subgraphs that correspond to higher-order concepts. For example, Klein notes how the concept of a series of events causing another series events could be reasoned about to construct a simple triple expressing $implies(A, B)$, where A is a composite node containing the subgraph corresponding to the first series of events and B is a subgraph constructed accordingly for the second series [597, pp. 5–7]. I will revisit this aspect of the architecture in Section 4.2.3 when discussing the notion of selecting a subset of simulated material for surface expression.

Monolithic Murder Mysteries

Klein's architecture probably appears to be quite modular at this point. But while it certainly *supports* the authoring of very modular procedural content, the content that was actually authored for the murder-mystery generator lacks modularity, and in two particular ways. To explore this, let us consider the diagram for the system, shown in Figure 4.3, which was supplied by Marie-Laure Ryan in a fabulous 1987 paper [1065, p. 514]. For something to be modular, it must have two characteristics: it is made up of small pieces, and the pieces can be recombined to construct more complex higher-order structures. As I will explain momentarily, Klein's system illustrates how the procedural content authored for a simulation may lack modularity in either or both of these senses.

[40]It appears that Klein and Robert F. Simmons, while working together at System Development Corporation (the first software company [89]) in the early 1960s, independently invented the notion of a semantic network [604], though they were preceded in this by the earlier projects of Richard H. Richens [999] and Margaret Masterman [755, 1184].

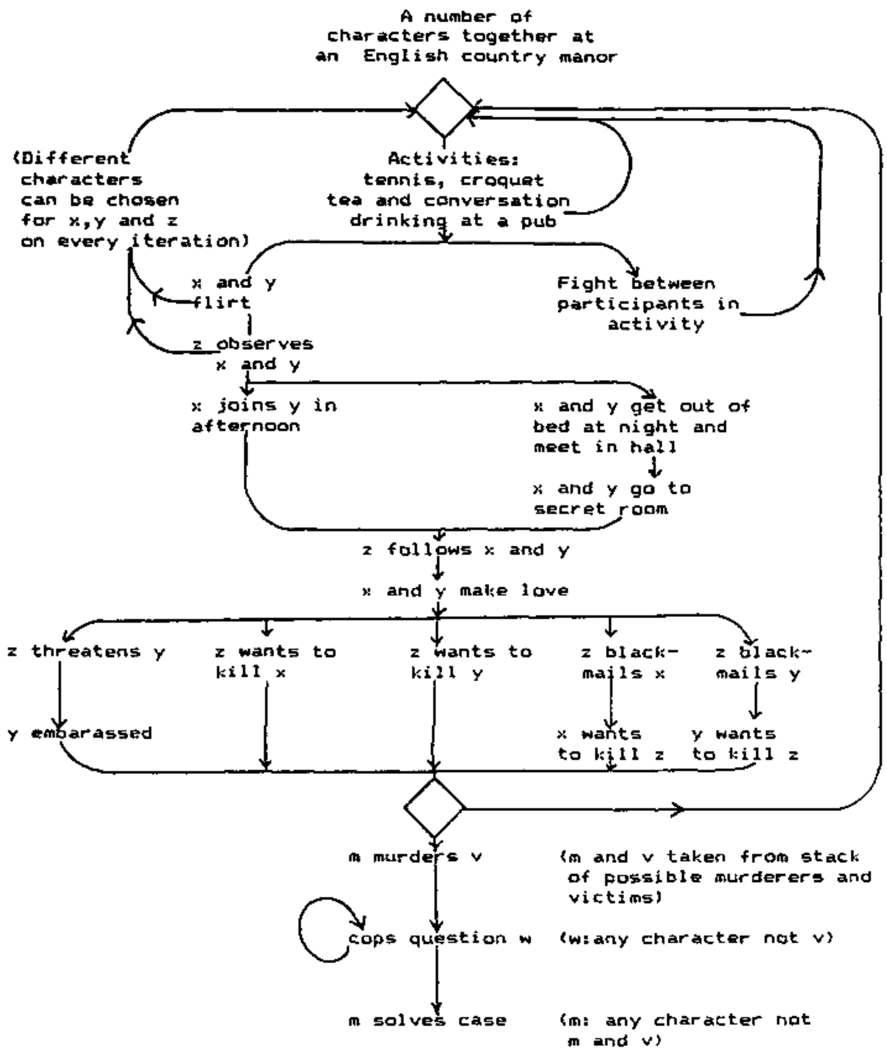


Figure 4.3: An illustration of emergent narrative in Sheldon Klein's murder-mystery generator. This diagram by Marie-Laure Ryan [1065, p. 514] illustrates the simulated storyworld developed by Sheldon Klein and his students in the early 1970s. To be clear, this visualizes the system's high-level narrative possibility space, not its workflow—the space depicted here is in fact yielded by a complex production system that models character behavior at the level of fine-grained rules. Further, the diagram simplifies in places by collapsing multiple possibilities into composite nodes (e.g., the murder node should branch into the six authored *modus operandi*).

Ryan's diagram shows that the procedural content that governs character behavior is made up of pieces, but some are more monolithic than others.[41] For example, while the modeling of character flirtation and a subsequent tryst spans multiple high-level events that may themselves recombine to yield a fairly rich possibility space, the questioning by the cops is a single monolithic node that feeds back onto itself in an act of repetition. A more modular approach would also break down the police investigation into its constituent parts, ideally in such a way that the parts could be recombined to yield meaningful narrative variation. One possibility, however, is that the program is meant to model the possibility space of an existing fiction genre, the murder mystery, and the tropes of that genre imply that it is not important to richly model (in this case) the nuanced possibilities of a bungling police investigation. This is a fair argument—and the particular monolithism under consideration here may also have been due to a lack of authoring resources, as Section 4.1.5 notes—but I believe that the authors of simulation engines for emergent narrative should strive toward modularity, since that yields greater emergence, which amplifies the aesthetic qualities of the form as outlined in the previous chapter.

The other component of modularity, besides breaking monolithic chunks into smaller pieces, is the capacity for those pieces to be recombined into a variety of higher-order structures. In Klein's system, we find cases where potential narrative beats do not branch out to enable one or more possible ramifications, but instead feed forward (more or less deterministically) into another narrative beat. For ex-

[41] As I also note in the figure caption, this diagram is not consistent in the fidelity with which it represents portions of the narrative possibility space. Specifically, while Ryan has taken care to represent all the major branch points preceding the murder, her depiction of the murder beat itself collapses a series of possibilities into one composite node. In fact, there are six ways that the murder can play out [597, p. 67], as determined by the particular motive (of which there are also six) that the murderer has taken on, which is itself determined by how the earlier simulation phases turned out. In general, Ryan's diagram does not fully capture the degree of coordination between emergent actions and later ones. Later, in Section 4.1.6, I discuss this notion of simulation feedback as perhaps the most critical design concern in emergent narrative.

ample, in the diagram shown in Figure 4.3, this feedforward pattern occurs in the linking of the nodes marked `x and y get out of bed at night and meet in hall` and `x and y go to secret room`. When a sequence of chunks is connected in this way, like beads on a string, the monolithic structure that was broken into these chunks is reconstituted—it has simply become a monolithic sequence, rather than a chunk. A more extreme example of this is the node called `cops question w`, which feeds back onto itself to produce the monolithic sequence of the same event recurring over and over again.

A Combinatorial Magic

Again, modularity is critical when it comes to the procedural content of emergent narrative, because it is what enables the rich possibility spaces that undergird the aesthetics of the form. A shining example of modularity in emergent narrative is the game *Caves of Qud*, by Jason Grinblat and Brian Bucklew [428, 427]. In a recent book chapter, Jason Grinblat has articulated the appeal of modularity:

> Modularity is the use of discrete units, called modules, to assemble larger structures, which we'll call gestalts. Our motivation for investigating modular design comes from the combinatorial magic that modularity conjures. We design a few modules, along with an assembly mechanism, and we inherit a plethora of gestalts for free. And gestalts are what matter. They're the artifacts our players encounter and care about, be they dynamic puzzles, dungeon levels, or dialog trees. Modularity multiplies our work, often bearing novel results in the process. [425, p. 29]

Tarn Adams has likewise endorsed the virtues of modularity, using an example from the development of *Dwarf Fortress*:

> If you just focus on the end result, it's not clear what's required to make it look and work correctly. Instead, break down and understand the system you are modeling in terms of its basic elements and interactions. Not only will you develop a richer interplay of objects, but

> certain problems solve themselves. As a simple example, when creating terrain, it is tempting to spawn particular biomes or allow a fractal to directly define the biomes. However, Dwarf Fortress achieved much better results by handling fields separately: temperature, rainfall, elevation, drainage, etc. The interplay of those fields determined the final biome, resulting in a more natural, internally consistent solution. [15, p. 520]

In total, the case study of Klein's system provided in this chapter shows that a modular architecture, though probably necessary for strong emergent narrative, is only half the battle: the procedural content authored for that architecture must itself be modular as well.

4.1.4 Lack of Abstraction

While modularity pertains to the granularity and combinability of procedural content, a related issue concerns the level of abstraction that is used to define such content. Here, let us return to Noah Wardrip-Fruin's critique of Klein's system, in which he compares it to *Tale-Spin* in a particular way:

> But the field has generally followed Meehan's critique of Klein's system: it was made up of explicit chunks of action, with a path through these chunks selected randomly. So, for example, Klein's system includes a 'rule for people arriving at George's living room'. Meehan's system, on the other hand, contains general rules for character movement (and reasons for movement) that operate no matter which spaces are available—living rooms, bedrooms, caves, or meadows. [1311, p. 116].

Here, Wardrip-Fruin is articulating a distinction between the two systems in terms of *abstraction*: he argues that Klein's system utilizes concrete chunks of procedural content that are explicitly associated with certain characters and places, while *Tale-Spin*'s procedural content is defined abstractly so that it may be utilized for any character in potentially any context.

Klein's Concrete Elements

As the discussion in Section 4.1.3 illustrates, Klein's procedural content does in fact utilize abstraction, with rules whose predicates may match against arbitrary features of the storyworld (and even aspects of the architecture itself, as a form of computational reflection). To be clear, though, there are also a few rules that reference specific concrete entities. In the example that Wardrip-Fruin cites, a rule determines which characters the host George will invite to his cocktail party [601, pp. 14–15]. This is not, however, a concrete plot event that is simply triggered by the rule. Rather, whom George will invite depends on aspects of the storyworld as defined in the initial state, namely his affinities for the other characters, and so changing the initial state may cause this rule to be triggered differently, which may totally change the story.[42] The reason George is explicitly referenced in the rule is that the setting of the initial murder-mystery domain (described in a preliminary report on the project) is a cocktail party at George's house, and so every generated story begins with George inviting his guests [601, p. 27]. Likewise, Klein's revised story domain (referenced in all subsequent reports on the project) also has a concrete setting—Lady Buxley's countryside manor [597]—and some concrete rules pertaining to specific characters and locations within the manor.

Thus, while Klein's system features abstraction, it also features concrete elements, and this is what underpins the fundamental critiques of the system (though, as I have shown, these are largely misguided).[43]

[42]In this sense, Klein's system actually works a lot like *Tale-Spin*, which Meehan described as being interactive due to its user affordances for asserting facts about the storyworld [823].

[43]*Tale-Spin*'s storyworlds also have concrete characters and settings at runtime, since generally these elements are defined at authoring time, just as in Klein's project. The major difference is that *Tale-Spin*'s interactive mode cleverly interleaves story authoring and story execution by asking the interactor to define aspects of the storyworld (e.g., one character's affinity for another) as they are needed to evaluate conditions on activated planning operators. In 1986, Laxmi Gupta amplified this aspect of *Tale-Spin* in a reimplementation of the system [435]. Later still, Michael Gardner further extended this interaction pattern to build a mixed-initiative authoring tool called *Multistory* [381]. As I write this, Ben Samuel is developing a system called

The Aesthetics of Abstraction

Note that Klein and his students could have replaced the concrete elements with more abstract procedural content. In this configuration, the stories would not all take place at the same setting—George's house in the first domain, and Lady Buxley's manor in the second—but instead rules would be authored for generating a setting. Likewise, the characters in the stories would not be the same every time, but instead they would be generated according to a different set of authored rules.[44] In this way, the story generator would become *roguelike*, and thereby the aesthetics of emergent narrative, which I outlined in Section 3.2, would be amplified. Let me explain how.

When setting and characters are concretely modeled, every generated story is about the same people in the same places, and the result may feel more like

Writing Buddy that pushes the idea even farther [1086]. Another related prospect for procedural narrative is the system itself filling in these deferred gaps automatically, in a way that would best serve the emerging narrative. This approach was explored by another early system by Masoud Yazdani, called *Roald* [1366]. More recently, is has been articulated as *late commitment*, a technique by Ivo Swartjes and his collaborators that takes its inspiration from improvisational acting [1231, 1232, 1229]. As I will argue at some length in Chapter 5, I find these approaches to be highly intriguing, but I believe they represent a departure from emergent narrative and its core aesthetics.

[44]As noted above, *Tale-Spin* also features concrete storyworlds and characters, but some of the authoring of such entities is offloaded to a human interactor. What would it look like if the system itself generated such material? Since *Tale-Spin* is all about planning, such a reformulation would probably use the same approach for this task. But while *Tale-Spin* only used planning for character behavior, this would constitute its utilization for *author* behavior. This is the basis for *Author*, a follow-up system that was proposed by Meehan's labmate Natalie Dehn [252, 249, 251]. Dehn's ambitious project developed alongside her advisor Roger Schank's theory of *dynamic memory* [1104], but *Author* does not appear to have been implemented [251]. (I have also gathered this from a phone call with Roger Schank, who did not recall the system being completed [personal communication, June 28, 2018].) Meanwhile, this intellectual period of the Yale AI Project also produced some of the earliest work on *case-based reasoning*, namely Janet Kolodner's *Cyrus* [1106, 613] and Michael Lebowitz's *IPP* [662, 664]. Shortly thereafter, Lebowitz developed *Universe*, an early example of how the integration of authorial and character planning might work in a story generator [663, 665, 666, 667]. Though conventionally unrecognized, this territory was actually first explored by Masoud Yazdani's little-known *Roald* system [1363, 1364, 1365, 1366, 1367], as Lebowitz himself acknowledges via citation [666, p. 485]. Later, Erik Mueller and Scott Turner—both students of Michael Dyer [288], who was himself a student of Schank's student Wendy Lehnert [289]—produced two other important systems in this tradition, *Daydreamer* [863, 862, 861] and *Minstrel* [1275, 1277, 1276].

fiction than nonfiction. Moreover, attendant to this kind of implementation may be a sense that with each newly encountered story an interactor is converging on a fuller understanding of a single monolithic, though multifaceted, story. These sensations are in direct opposition to the pleasures of emergent narrative, and as a result such concreteness works to inhibit its core aesthetics. By authoring characters, the aesthetics of the unauthored are damaged, and in featuring the same characters across everyone's stories, the aesthetics of the personal are undermined. One component of the aesthetics of the vast is the magnitude of the possibility space not just within a given simulated storyworld, but across all the storyworlds that a simulation may produce—because this higher-order possibility space is constrained when concrete characters and settings are used, the aesthetics of the vast are inhibited in turn. Finally, when a reader or interactor finds that the characters of her world persist across simulation instances, the aesthetics of the ephemeral are diminished.

Along the Abstraction Continuum

In procedural narrative, the extreme case of low abstraction is probably represented by something like Arthur Blanchard's *Movie-Writer* [118], which is the earliest project in the area of which I am currently aware. Patented in 1916, *Movie-Writer* is a small device that enables the semiautomatic production of outlines for silent-era movie scenarios. As Figure 4.4 illustrates, this works by a simple process of recombination, where concrete chunks of content (each being a single word) are mechanically juxtaposed.[45] A later 1930s project by Wycliffe A. Hill

[45]Related mechanical devices for procedural generation include Henrietta Rosa Montague's *Name Selector* (1947) [1026] and Norvell E. Von Behren's device for use in baby naming (1968/1971) [1298, 1299]. For a history of procedural generation in analogue games, see Gillian Smith's fabulous paper [1172]. Moving back to procedural narrative, we might consider the most extreme possible case of concreteness. To my imagination, this would be something like a rolodex of *complete* stories, the 'procedure' of which would be actuated by a human shuffling

works on the same principal by allowing interactors to generate storylines using a device called the *Plot Robot* (alternatively, the *Plot Genie*)—a paper number wheel that works as a pseudorandom number generator—to recombine character and event descriptions found in a series of books [484, 485, 486, 487].

Moving toward the other side of this continuum, we find *The Uranium Shipment and the Space Pirates* (1955), an extremely interesting artifact that Brian Moriarty has identified as the earliest known work of digital interactive storytelling [856]. Specifically, it an example wiring of the *Geniac* kit [102] that is included in its instruction manual [382, pp. 30–31]. Marketed as "an exciting interplanetary chase pitting yourself against the machine" [1, p. 16], this *Geniac* wiring allows a human to select initial conditions for a procedural sci-fi story by turning a set of discs on the face of the kit. By the power of combinational logic, each input configuration automatically triggers a particular ending to the story, which is indicated to the interactor by a lightbulb display.[46]

Further along the abstraction continuum, we encounter *story grammars* [1034]—

through the rolodex (or simply revealing the next card). Here, there is no recombination, or even any generation at all, but instead a pseudorandom *selection* of complete stories. In fact, this rolodex would be procedurally equivalent to a book of short stories that a reader may page through pseudorandomly, or even in order. As such, this case does not even feel like procedural narrative. Indeed, it seems that some amount of abstraction is necessary for the form to even obtain; in the case of *Movie-Writer*, such abstraction comes in the form of its word groupings and the scenario template that houses them in an ordered array of slots. *Movie-Writer* feels generative, but it could also be seen as an example of selection: the wheel positions index the stories, and through turning its wheels, one of its 200^6 stories is revealed. Why does it feel generative when the stories are broken into chunks? Indeed, all procedural generation may be viewed as being essentially a matter of selection—that is, selection of one artifact from the generator's possibility space (this is the sense evoked by the AI term of art *search* [614, 1259]). From this perspective, all a system's generable outputs exist all along in the space, and through some procedure—computational, human-driven, or hybrid—a particular output is *retrieved*. Taken to its extreme, this notion could lead us to view computation itself as a matter of retrieval: a computing machine is a device for retrieving data instances that already exist in the space of possible data that is yielded by the device's electromechanical system. I return to this ontological position in footnote 87 of Chapter 4.

[46]Moriarty's student Zackery Mason has developed *Virtual Geniac* [754], a virtual simulation of the *Geniac* device that allows one to experience of *The Uranium Shipment and the Space Pirates* today.

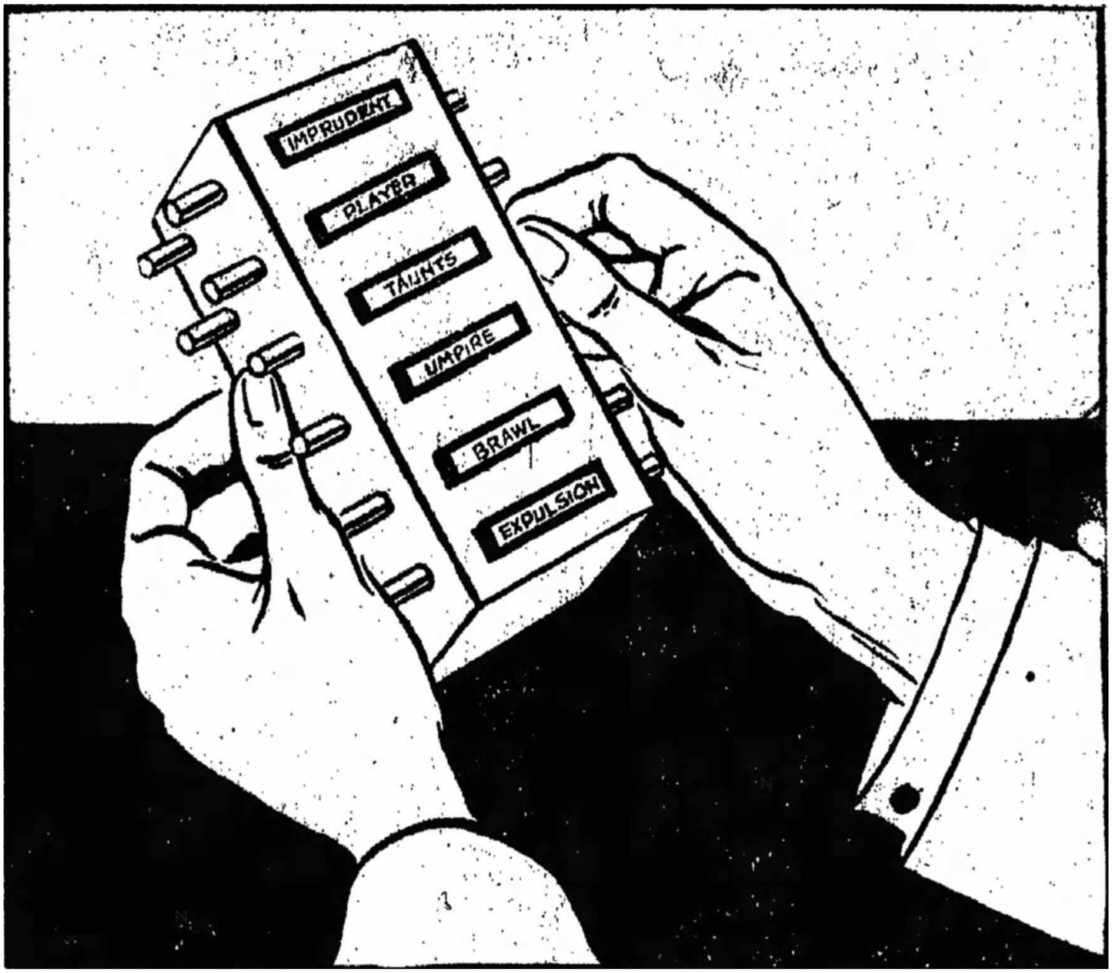


Figure 4.4: An illustration of Arthur Blanchard's *Movie-Writer*, a little-known mechanical story generator that was patented in 1916. The device is a small wooden box containing six paper rolls, each of which is wound around a pair of spindles. Each paper roll corresponds to a part of speech and has 200 words belonging to that category printed on it; selected words can be seen through slots in the wooden box. By turning the spindles of a particular roll, the interactor may cause new words to appear in the corresponding slot, and turning multiple spindles quickly will cause a new recombination of six words to appear semiautomatically. Critically, the six word categories are constructed such that any recombinant string of words may be read as a rough outline or pitch for a silent-era movie scenario. An interesting early example of procedural narrative, *Movie-Writer* represents perhaps the least amount of abstraction possible in procedural narrative.

and well-documented antecedents such as Propp's folktale model [960] and William Wallace Cook's *Plotto* [212], as well as several obscure earlier ones [690, pp. 171–189][950, 483, 914]—where abstraction appears in the form of nonterminal symbols (representing plot events or other story material) that may be rewritten in different ways depending on the grammar's production rules. Moving further still, we find *roguelike* configurations, where characters and other story elements are procedurally generated for each runtime instance. Examples of such a configuration include *Dwarf Fortress* [17], *RimWorld* [1209], *Caves of Qud* [428, 427], and my own simulation engines *World*, *Talk of the Town*, and *Hennepin*, which I describe at length in Part II of this dissertation.

To produce something at the far end of this continuum—an antithesis to the approach of *Movie-Writer*—would probably require something like *program synthesis*, whereby the procedural content that defines the rules of the simulation would itself be generated [736].[47] Steps toward this include the synthetic behavior trees of *Façade* [777] and the modular entity system of *Caves of Qud* [425].

Retargeting in *Comme il Faut* and Klein's System

I would like to end this section by discussing a particular method of abstraction in procedural narrative, and in software development more broadly, called *retargeting*. In character animation, this term refers to a technique whereby abstracted animation patterns may be authored independently of any given character rig and

[47]Perhaps this is what Sheldon Klein was working toward with his increasing exploration into computational reflection. While the most famous work on program synthesis occurred in the mid-1980s [735], it has a much longer history. Indeed, the task is sometimes called *Church's problem* [1248], due to Alonzo Church's 1957 proposal for automatic circuit synthesis [188]. Even earlier, in 1954, the term 'automatic programming' was applied to early Fortran compilers, since at that compilation was seen as a kind of program synthesis [71, p. 1257]. When Harrison "Dit" Morse undertook development work on *Saga II* in 1960, he developed a series of *macrocode* layers that would each compile code at one level into lower-level code at the next [1183, 3m0s]. At the time, this work was at the cutting edge of software engineering practice.

then dynamically retargeted to a specific rig at runtime [394]. While the notion originated in computer graphics, it may be more generally conceived as a software design pattern [373], and there is a clear application to procedural narrative: procedural content, such as character actions, that may be retargeted to various characters or other storyworld entities. Here, it appears that the term was first applied to this domain by the creators of *Prom Week* [799, 803], or more precisely, its underlying *Comme il Faut* (*CiF*) architecture [804]. As Michael Mateas and Josh McCoy explain in a book chapter dedicated to the topic, this is how retargeting works in *CiF*:

> Multicharacter social exchanges are explicitly represented separately from any specific characters. Given a cast of characters, with traits and social state declaratively represented, social exchanges are retargeted for specific characters. [770, p. 517]

In *Prom Week*, this means that both character *social exchanges* (actions that cast multiple characters in specific roles [804, p. 100]) and their *instantiations* (concrete performances of those exchanges that couple natural-language dialogue and character animations [804, pp. 100–101]) may be retargeted to any character. To enable such retargeting, an author defines (using condition logic) the social and narrative patterns in which the procedural content may be used, as well as the rules for rendering concrete instantiations of that content for a given set of characters (or, more broadly, for a given emergent context). Note that Klein's system also features this kind of retargeting, with actions that may be taken by variant characters pending dynamic aspects of the storyworld. Interestingly, Klein and his students intended for the "preselected hero" Dr. Hume to always be the character who solves the mystery [597, p. 70], but this is not possible in all storyworlds: depending on his personality and affinities for other characters, Dr. Hume may end up becoming either the murderer or the victim. When this happens, the procedural content by which the crime is solved will be retargeted to another randomly selected character [597, p. 70].

Roguelike Retargeting

There is a more extreme case of retargeting than that which is utilized in *Prom Week* and Klein's system: instead of retargeting behavior to any preauthored character, behavior may be targeted to any *possible* character. This is the roguelike approach to retargeting, and it is at play in projects such as *The Sims* [792], *Dwarf Fortress* [17], *Caves of Qud* [428, 425], and my simulation engines that are discussed in Part II of this dissertation. As I explain in Chapter 11, in authoring procedural content for my most recent system, *Hennepin*, I conceptualize narrative systems like the one developed for Klein's project as *story machines* that may be retargeted to arbitrary characters at arbitrary times and places. Indeed, retargeting may be conceptualized at different levels of abstraction, and at different levels of hierarchy, with regard to a system's procedural content.

Authorial Leverage and Authorial Affordances

Generally, retargeting—and the broader notion of abstraction—is useful in emergent narrative because it provides both *authorial leverage* and *authorial affordances*. Authorial leverage is a concept that originates in a 2009 paper by Sherol Chen and collaborators, and it refers to the ease with which procedural content may be authored in a given AI architecture [180]. More specifically, when authorial leverage is higher, each unit of authoring effort counts for more. Quoting from an unpublished manuscript, Chen's collaborator Mark Nelson defined it in this way in a recent personal communication:

> The 'authorial leverage' of a system is the effort-multiplier the system gives the author, i.e. the ratio between the effort it takes to author

> within a particular AI system and the effort it would take to achieve the same desired effect 'manually'.[48]

In the case of a system with concrete authored characters, retargeting provides authorial leverage by making authoring effort apply to multiple characters at once, as Mateas and McCoy argue [770, p. 516]. When a system takes the roguelike approach to character authoring (unique ones are generated for each storyworld), retargeting actually becomes a necessity. Without it, an author would have to take the drastic means of authoring procedural content for every possible character, but that is a ridiculous prospect.

The notion of authorial affordances is rooted in the work of Michael Mateas [766], who defines them as "the 'hooks' that an architecture provides for an artist to inscribe their authorial intention in the machine" [764, p. 125]. Retargeting provides such a hook by allowing an author to reason about abstract patterns of behavior that are independent of any concrete characters. This affordance is critical to sculpting the massive possibility spaces of emergent narrative, since it facilitates both the defining and the understanding of such spaces. When it comes to inscribing one's intent into a system, it is critical that the author can not only express her intent, but also understand how the authored material reifies that intent. Abstraction makes authoring more feasible because it reduces the combinatorial authorial burden that would be incurred by enumerating all possible concrete content configurations. In turn, abstraction allows an author to reason more effectively about a system, because it reduces the massive conceptual space of possible content configurations into a smaller abstracted space.

[48] Personal communication, May 4, 2018.

4.1.5 Modeling Gaps

Simulations have gaps in terms of what they model. That is, all simulations fail to perfectly model the thing they are simulating—this is core to the very definition of simulation, and were the case otherwise for a given system, it would not be a simulation but rather the thing itself.[49] Often, what is modeled in a simulation may call attention to what is not modeled but could reasonably be expected to be, given what is already included. I call these *modeling gaps*.

Agency Issues as Modeling Gaps

In a sense, this idea relates quite closely to the notion of *agency* as defined by Wardrip-Fruin and colleagues [1317, p. 1]:

> a phenomenon, involving both the game and the player, one that occurs when the actions players desire are among those they can take (and vice versa) as supported by an underlying computational model.

In an earlier paper on this sense of agency, Michael Mateas argues that *Quake* is an example of a game with high agency [763]. This may seem counterintuitive, because in *Quake* players cannot do much besides moving around and shooting at things. Critically, however, the gameworld as it is modeled does not suggest the possibility of taking other actions; no player is shocked to discover that a Death Knight cannot be engaged in conversation, for instance.[50] As such, this articulation of agency is about a kind of modeling gap, specifically one that pertains specifically to the affordances that are available to an interactor in an interactive work. The connection to the kind of modeling gap that I am outlining here relies

[49]I should note that it is possible to build computer programs that engage the aesthetic practice of simulation without actually modeling a subject phenomenon in the real world—Conway's *Game of Life* [380] is a classic example—but in the case of emergent narrative a simulated storyworld will likely model real phenomena to some degree.

[50]This would make for an interesting mod, though, for that very reason.

on the stipulation that a gap may only be identified relative to what is actually modeled. Thus, while the very ontology of simulation depends on modeling gaps, the stuff that is modeled may call particular attention to what is not modeled.[51]

Klein's Modeling Gap: No Detection

An example of this phenomenon occurs in the murder-mystery generator developed by Sheldon Klein and his students. Let us consider an excerpt from "Murder Mystery 1", one of the outputs generated by the system. This segment occurs after the murder has been discovered:

> The cops arrived. The cops were idiotic. A detective examined the corpse. The policemen looked for hints in the bathroom. Dr. Bartholomew Hume also looked. Edward tried to calm Marion. The policemen questions Dr. Bartholomew Hume. The detective asked questions. The policemen searched the garden. The policemen tried to find clues. Marion cried. Dr. Bartholomew Hume searched stairs. Hume looked for hints. Dr. Hume questioned Lady Buxley. Dr. Hume knew that Lady Buxley told the truth. Florence talked with Heather about the murder. Marion cried. The policemen questioned Ronald. The inspector suspected Ronald. The inspector asked the stupid questions. The policeman searched the parlor. The policemen tried to find hints. Florence was upset. Dr. Bartholomew Hume searched the dining room. Dr. Bartholomew Hume looked for hints. The cops questioned Heather. The detective asked the stupid questions. Dr. Hume questioned Heather. Dr. Hume knew that Heather told the truth. The cops searched the tennis court. Clive talked with Ronald about the murder. The butler said James was kind. The cook talked about the murder. Dr. Bartholomew Hume searched the bathroom. Dr. Hume looked for clues. Marion cried. Dr. Hume questioned Florence. Hume knew that Florence told the truth. Dr. Bartholomew Hume got information from Florence. The cops searched the bathroom. The cops found a thread. The thread was a misleading clue. Lady Buxley talked with John about the murder. Lady Buxley said that James was kind. Dr. Hume was upset. Dr. Bartholomew Hume searched

[51] Note that modeling gaps are in the eye of the beholder. That is, the degree to which a gap is noticeable will vary across individuals, since cultural contexts and other considerations frame the interpretation of a simulation's modeling.

> the library. The cops questioned John Buxley. The detective asked the stupid questions. Hume questioned the cook. Dr. Bartholomew Hume knew that Maggie told the truth. Hume got information from the cook. Hume went to the bathroom. Dr. Hume found the bottle. Hume knew the murderer. Hume asked everyone to go to the parlor. Dr. Bartholomew Hume said that the murderer was in the room. Everyone was surprised. Everyone talked. Dr. Bartholomew Hume said that James was killed by poison. Hume said that the butler killed James. Everyone was shocked. [597, pp. 104–107]

This system models the genre of detective fiction, and in particular the murder mystery. One trope in this genre that is clearly targeted by Klein's system is one in which a lay character, amid a cast of bumbling law enforcement personnel, solves the crime through clever deduction. While Dr. Bartholomew Hume does appear to do some detection—he searches rooms and questions some of the other characters—there is no running thread that connects these actions into a process of deduction. A hallmark of detective fiction is the detective's deductive process that culminates in a detailed account of the circumstances of the crime, but it is missing here.[52] Because the system targets detective fiction, and because a number of other tropes of that genre are actually modeled—countryside manor, trysts, a murder, multiple character with motives, bumbling law enforcement, amateur detective, etc.—the lack of a deduction simulation is a modeling gap.

In what it is currently to my knowledge the earliest dissertation in all of computational narrative, *Foundations of a Computable Theory of Narrative* (1973), Ed Kahn provides an extensive critique of Klein's system.[53] As a core part of

[52]As Ian Horswill noted in his feedback on an earlier draft of this thesis, this feature is less activated in certain traditions, namely the American *hardboiled* genre [173]. Likewise, in his own comments on the draft, Noah Wardrip-Fruin mentioned an alternative pattern that has been called the *reverse whodunnit* [1280]: the audience knows all the details upfront, but suspense obtains through uncertainty as to whether the detective will be able to reach the proper solution. Nonetheless, I think Ian and Noah would agree that Klein's system misses the mark.

[53]Note that Kahn analyzes "Murder Mystery 1" but calls it "Sheldon's Revenge": "In the story given in Appendix 1 which I have taken the liberty of referring to as "Sheldon's Revenge," "oversexed Marion" establishes several liaisons." [559, pp. 14–15]. This was perhaps meant

this critique, Kahn identifies the modeling gap under discussion as the system's fundamental flaw:

> The essential thing lacking in a story such as "Sheldon's Revenge" is any very substantial realization of the detective function. It is not a detective story because there is no detection. Dr. Hume announces that he knows the killer, but never says how this conclusion was reached. There should be an episode of explanation in which the relevant clues are presented. [559, p. 18]

Klein, believing that aspects of Kahn's critique were rooted in a misunderstanding of the project, initiated a public dialogue that was published in the April 1974 SIGART Newsletter under the heading "Automatic Novel Writing—An Exchange."[54] In the exchange, Kahn characterizes the problem of the deduction modeling gap in this way:

> Stories such as MM1 fail to be detective stories because there is no real detection. There is no purposeful acquisition of knowledge on the part of Dr. Hume, he simply announces that the butler, Clive, did it. Real detective fiction involves a step by step enactment of the deduction process. Instead we get a plain announcement, based upon no explicit argument whatever. This failure of be detective fiction is syntactic, the necessary episodes leading to capture are missing. [558, p. 3]

In his dissertation, Kahn proposes an interesting solution to this modeling gap, in the form of a detective simulation that would utilize Klein's notion of a "private

---

playfully, but in a phone call last year, Kahn told me that he regretted doing this, especially since it upset Klein (personal communication, June 12, 2017).

[54] The crux of Klein's rebuttal is that Kahn conflated Klein's architecture with the procedural data that was authored for it (as a model of the murder-mystery domain). That is, what were in fact expressions of deficiencies in this authored data were misapprehended by Kahn as being expressions of fundamental limitations in the architecture. In fact, Klein's architecture, which was later christened the *Meta-Symbolic Simulation System* (*Messy*) [46], is in my estimation an extremely powerful (and elegantly designed) simulation engine. To name a few examples that are familiar to me, this conflation would be like attributing deficiencies in a particular *Tracery*-fueled *Twitter bot* [638] to *Tracery* itself [205], or those of *Prom Week* to *Comme il Faut* [804], or those of *Façade* to *ABL* [778]. I believe that this phenomenon contributes to the misunderstanding of all sorts of computational systems, particularly generative ones in which the architecture and domain model are decoupled. I propose to call it the *Messy effect*, following a related phenomenon of human misunderstanding, Noah Wardrip-Fruin's *Tale-Spin effect* [1310, 1311].

semantic universe", which is a mental simulation by which a character recursively runs the storyworld simulation in her mind (with an initial state corresponding to her subjective beliefs about the world) [597, pp. 29–30].[55] Here is Kahn's proposal, as articulated in his dissertation:

> It is worthwhile trying to extend Klein's plot specification rules to see what is needed to fulfill the detective function adequately. If Dr. Hume had access to the deep structure of the narrative universe, i.e. its semantics, he would sooner or later observe that Clive was a poor relative of the wealthy victim. If furthermore Dr. Hume had a private semantic universe in which murder motives could be computed and being a poor relative was one such motive, then Dr. Hume would have reason to suspect Clive. Clive's access to poison would be an inference based on his position as a servant in a large house. These three components, a motive model, a means of gathering information for that model, and knowledge about access to weapons, seem minimal to establish some plausible basis for Dr. Hume's otherwise occult source of knowledge. [559, pp. 18–19]

In a refreshing moment of candor that could only appear extraneously to formal academic writing, Klein's rebuttal acknowledges the modeling gap and provides a simple explanation for its existence:

> Kahn is right in saying that, in this particular detective story model, the true detective function is absent. This defect is an artifact of the need to construct an example quickly in time for a conference presentation, and in time for my immediately subsequent visit to Moscow on an exchange program. [558, p. 4]

[55]Klein's description of this technique is quite intriguing. The implementation plan allows a character to literally run a recursive instance of the simulation code, which means that a storyworld (and its evolution over time) truly obtains in the character's mind. For one, Klein notes, this could allow for planning by subjective look-ahead: "An individual character could be made to resort to his own look-ahead simulation of events in order to evaluate decision making criteria about the implication of current actions on future events" [597, p. 19]. More evocatively still, Klein also suggests that the approach could be used to implement character dreams, theory of mind (by running a recursive simulation of another character's mind, given beliefs about their beliefs), and time travel. As Noah Wardrip-Fruin noted in his feedback on an earlier draft of this thesis, Klein's private semantic universes remind him of the way in which *Tale-Spin* [822] characters plan ahead by unraveling an extensive series of possible futures. I will discuss this phenomenon, and Noah's own writing about it [1311, pp. 147–151], in Section 4.2.6.

Are Modeling Gaps Political Statements?

In the 1987 edition of his seminal *Computer Lib/Dream Machines*, Ted Nelson writes the following about computer simulation (under the header "All Simulation is Political"):

> Every simulation program, and thus every simulation, has a *point of view*. Just like a statement in words about the world, it is a *model* of how things are, with its own implicit emphases: it highlights some things, omits others, and always simplifies. [876, p. CL149][56]

In a recent article, Noah Wardrip-Fruin expands on Nelson's statement:

> all the simulations created with these [simulation authoring environments] embed assumptions about the world that derive from viewpoints—they are political. [...] In an individual simulation we see the politics in the rules and data. [1315, p. 108]

Additionally, recent papers by Peter Mawhorter [784] and Kate Compton [203] have expounded on this ideas, as have others with particular application to procedural generation [941, 1174, 209].

I think there is some additional nuance, which I would like to discuss here, pertaining to the practical aspects of simulation crafting as a creative practice that

[56] It does not require a huge leap to extrapolate Nelson's point as being about all computer programs, not just simulations. Indeed, even seemingly innocuous programs like database systems can be highly political [500]. However, one kind of program that is sometimes viewed as being in a sense unauthored, and thereby free of bias (since there is no explicit creator who inscribes bias into the program), is a neural network. This is not the case, however, since neural networks evolve (*are* evolved) through the use of training data, which will almost invariably encode biases [1362]—*bias in, bias out.* Moreover, as Michael Mateas argues, building and understanding such a model entails subjective human interpretation: "These approaches still require the interpretation of an observer in order to make sense of the input/output relationships exhibited by the system, to select the primitive categories (features) with which the inputs are structured, and to tell stories about the processes producing the input/output relationships. These stories are essential for thinking through which technical constructions to try next, that is, for simultaneously defining a notion of progress and a collection of incremental technical constructions that make progress according to this notion" [765, p. 61]. Fortunately, the ethical and social dimensions of machine learning are the subject of a growing research area that aims to raise public awareness of these issues [132, 147, 1362, 148].

operates under constraints that ultimately affect modeling. First, computer simulation is constrained by hardware limitations that create computational memory and speed issues; this makes it difficult to simulate complex phenomena. Thus, even if an author has a particular theory about how a subject phenomenon works, hardware limitations may prevent her from operationalizing that theory in a computer simulation. Under such constraints, the resulting simulation may highlight, omit, and simplify in ways that make the simulation deviate from her theory, and thus the program's point of view is not hers. Sometimes an overly simplified model is the only one that is feasible to build.[57]

Further, simulation crafting, like any creative practice, has an associated set of skills that, when not honed in a particular creator, will contribute to limitations in that creator's capacity to express herself in that medium. Thus, if I am not particularly good at simulating, it will be hard to express myself through my simulations, and so my viewpoints may not be adequately captured by them. While we understand this more naturally in other mediums—no one thinks a bad painter who unintentionally botches her depiction of a tree is meaning to express a peculiar idea of what a tree is—the notion of skill as a factor in simulation crafting is not often acknowledged (in my experience). It is hard to adequately model complex phenomena!

Relatedly, while stylistic approaches that might be called 'impressionist' or 'expressionist' are in the case of other mediums understood to be just that—*stylistic approaches*—simulations are typically interpreted as attempts at literal representation (i.e., 'realist' or 'naturalist' expressions).[58] As an example, Anne-

[57]Interestingly, computationally simpler models sometimes correspond to the simplifications that undergird belief systems that are viewed as problematic [500]. Perhaps the cognitive simplifications result from a similar kind of computational limitation.

[58]While I do not know of examples in the area of world simulation, there are text generators that are driven by what I would deem to be impressionist and expressionist inclinations: respectively, Erik Mueller's *Daydreamer* [862] and Fox Harrell's *Griot*, particularly the latter's piece

Mette Albrechtslund, writing about gender representation in *The Sims 2*, equates simulation with an "aesthetical realism" [22, p. 1]. It is possible to express through computer simulation without attempting to literally represent a subject, and it is important that such efforts be interpreted accordingly. More troublingly still, simulations are often interpreted not only as attempts at literally representing how a subject is, but as moreover instantiating idealized views of how a subject *should be*. As an example, in her paper that I have just mentioned, Albrechtslund assumes that *The Sims 2* encodes the ideological position of its designers [22]. Some simulations may be created in this way, but this is not by default.

Now, I do not mean to claim that all or even most modeling gaps can be accounted for by the preceding justifications. I am compelled to articulate them, however, because in my personal experience with simulation crafting, I have encountered the corresponding misconceptions frequently. There is one last explanation, however, that I think probably does account for most modeling gaps: it takes a lot of time and energy to craft a simulation, and the work is never done, because true verisimilitude (pertaining to a subject phenomenon) can never be achieved (due to the ontology of simulation). It takes time to model anything in a simulation, and invariably there will be gaps that are never rectified. In the case of Sheldon Klein, we know by his own admission that his system fails to model a detection procedure (on behalf of the character who solves the crime) simply because his development team ran out of time and energy. While modeling gaps can indeed reflect a simulation creator's belief system—what she does not think is true (or important) about the subject phenomenon—they may also result from a simple lack of resources. For each of my own simulations, I maintain to-do lists that have grown to document hundreds of modeling gaps or other problems that I have identified and that I hope to rectify. The simple truth is that I will never

"The Girl with Skin of Haints and Seraphs" [452].

be able to fill all these gaps.

Before moving on, I would like to emphasize that a system modeling one thing and failing to model another does not necessarily mean that the former is viewed by its creator as being more important. Sometimes an author plucks the lowest-hanging fruits first, thereby saving the more challenging modeling gaps for future work. In the case of Klein's system, I wonder if there is no detection simulation in part because such a system would have to rely on the modeling of phenomena that precedes detection in a detective story—the establishment of criminal motives, a crime, discovery of the crime—and so it would not have made sense to model detection until the earlier subsystems were implemented. Klein did produce the antecedent systems, clearly, but unfortunately he never got all the way to detection.

Finally,

Modeling Gaps, Closure, and Apophenia

Finally, it is worth noting that interactors may be inclined, whether consciously or not, to fill in such modeling gaps as a matter of comprehending a simulation or a story. Indeed, even the densest story does not spell out its every detail, but instead the author assumes much about the reader's ability to connect the dots.[59] Writing specifically about the space between panels in comics, Scott McCloud refers to the broader phenomenon as *closure* [796, p. 63]. Apparently, this concept, as well as others found in McCloud's brilliant *Understanding Comics* [796], served as major inspiration for Will Wright as he was developing one of the classic works of emergent narrative, *The Sims* [355, 176]. In the case of simulation, human observers may be inclined to narrate accounts that postulate linkings between

[59]The profound inability of computers to fill in such gaps is the impetus for the branch of artificial intelligence called *commonsense reasoning* [243].

emergent events that were not actually linked in the underlying simulation—this is *apophenia* [1203] at work. In Chapter 10, I articulate a simulationist design pattern that I call *apophenia hacking*, whereby apophenia is deliberately targeted.

4.1.6 Causality Issues

Let us consider a classic distinction between story and plot, originally proffered by the writer E. M. Forster:

> "The king died, and then the queen died" is a story. "The kind died, and then the queen died of grief" is a plot. [350, p. 86]

Here, Forster is working to develop particular senses of 'story' and 'plot' as, respectively, narrative with explicit temporal relations and narrative that also has explicit causal relations. For our purposes here, however, we might simply consider this as the distinction between story and non-story, as others scholars of computational narrative have done [312, p. 44][p. 23][1072].

Temporal and Causal Relations

What is missing in Forster's first sentence is a relation between the two events beyond the order in which they occurred, that is, something beyond their *temporal relation*. While temporal relations among events are critical in narrative, such relations are not enough to make a story (their presence is a necessary condition, but not a sufficient one). Fortunately, temporal relations are a given in the kind of forward simulation that we find in emergent narrative—the system, omniscient over its modeled purview, knows the order in which its simulated events transpired—which precludes the potentially challenging task of pinning down this information (a task that is central in narrative forms such as investigative journalism and legal argumentation). But, again, temporal ordering alone is not enough:

this is the crux of my coadvisor Michael Mateas's critique of emergent narrative as *just one damn thing after another*.

The real juice in narrative is what separates Forster's second sentence from the first: *causality*. In the second sentence, we encounter a meaningful relation between the two events beyond that the first preceded the second,[60] which is that the king's death in fact *caused* the queen's death—put differently, there is a *causal relation* between the two events. This brings us to what I view as the most fundamental pitfall in emergent narrative: a lack of meaningful causality between the emergent events of a simulation.

The Dense Causal Structures of Life and Simulation

To be fair, simulations by their nature tend to be cybernetical: state evolves as a function of the previous state. In the case of emergent narrative, this means that character actions are typically taken in accordance with some aspects of the current simulation state. In Klein's system, discussed in Section 4.1.3, we saw this in the form of action preconditions that match against asserted facts in the working memory (which stores all the information constituted in the storyworld). Because actions update this working memory and subsequent actions reference it, we can think of the working memory as a mediator that ensures causal linking between events. That is, the causal mechanism is a chaining between the effects of earlier character actions and the preconditions of later actions.

The problem here is that this results in a kind of *massively distributed causality*: a given simulated event will likely be caused by a very large subset (or even all) of the events that preceded it. This occurs when a given action has preconditions

[60] This temporal relation, by the way, is expressed by the *discourse marker* 'then' [1112, pp. 248–266]. In my first project at UC Santa Cruz, I attempted to automatically discern the *temporal coherence* of blog stories (clarity of the true temporal relations among events) in part by reasoning about the usage of temporal discourse markers [1051].

that match the effects of many earlier actions, which means that the cause of the action can only be understood as being attributed to (or distributed across) all of the earlier actions. But this can get out of hand: the result of this kind of mechanism is a densely interconnected lattice that basically expresses that all earlier actions caused all intermediate actions to cause all later actions.[61]

The Narrow Causal Structures of Narrative

While the real world may work like this, stories tend not to. In narrative, we find narrower causal structures, with events that cause at most a few other events, and cases of convergence, where the forward causal flows of multiple events converge to cause a single later event. Topologically, these structures resemble sparse directed acyclic graphs, as opposed to the densely interconnected lattices of realistic (and simulative) causality.

Moreover, narrative causal structures tend to bind according to a specific kind of causal link: *contingency*. As the narratologist Mark Alan Powell explains, it is helpful to think about causality in terms of three kinds of causal relationships:

> To understand the plot of a narrative, it is also important to recognize elements of causality that link events to each other. Causal relationships between events may be subdivided into categories of *possibility*, *probability*, and *contingency*. Only in the latter case can one event actually be said to *cause* another. The first category refers to instances when an event simply makes possible the occurrence of another. Relationships of probability, likewise, are those in which one event makes the occurrence of another more likely. [955, p. 40][62]

[61]In his comments on an earlier draft of this thesis, Ian Horswill referenced a related phenomenon in robotics: "This is sort of like something that comes up in reactive robotics, where you have a bunch of independent behaviors that get triggered bottom-up. If you aren't careful, it will flit between unrelated actions rather than pursing a coherent course of action. That variously gets referred to as 'chattering', 'dithering' or 'lack of hysteresis'" (personal communication, July 28, 2018).

[62]Writing about probability in the context of philosophy in 1912, Abraham Wolf identifies these three kinds of causality in addition to a fourth one: *chance* [1348, pp. 329–330].

When a contingency relation linking two events is identified, the earlier event is understood as explicitly causing the later one. Rarely will an event have been contingently caused by more than one or a few earlier events. This is the kind of causality that predominates in narrative: in topological terms, sparse directed graphs whose edges represent contingency relations between events.[63]

Inscrutable Causality in *Prom Week*

Simulation feedback should result in what I view as the holy grail of emergent narrative: *emergent contingency structures*. Here, I refer to the kind of narrow causal structures that one finds in narrative, as opposed to the massively branching structures that characterize how causality actually works in the real world and in simulation. That is, an emergent contingency structure is bound exclusively by contingency relations between events, and such a sequence is an important one if it includes an important event.[64] A nice example of the distinction between this kind of causal structure and a more realistic one (capturing all three kinds of causality) comes in the form of *Prom Week* [799, 803].

As I explained in Section 4.1.2, in the game's underlying simulation, characters decide what they want to do by reasoning over the state of the storyworld according to thousands of *social considerations* [799]. Specifically, each of these considerations is represented by a production rule whose preconditions match against the simulation state and whose effects increment the utility of taking a certain action. So what is the cause of a *Prom Week* character action? It is con-

[63] The designer and thinker Bret Victor has argued (on Twitter) that the simplistic nature of causality in narrative actually makes it an inadequate medium for evoking empathy: "Narrative inherently embraces simple causes, personal agency, single perspective, limited context. A medium for empathy won't be stories" [1295, n.p.]. My labmate Max Kreminski has associated this stance with the catchphrase "stories considered harmful" (personal communication, July 27, 2018).

[64] Something akin to what Labov calls the *most reportable event*. As noted in Section 4.1.1, this is an event that may form the crux of a story.

stituted in all of the rules that fired to increase the utility of that action during the procedure for action selection. More precisely, an action's causality is rooted in all of the *social concerns* captured by all of the preconditions on all of the fired rules.[65] In each case, the specific causal mechanism that binds such concerns to the selected action is Powell's *probability* relation, since the fired rules work to make an action more likely. Moreover, a whole suite of other executed rules may have worked to *decrement* the utility of other actions that may have been taken instead (had those rules not been triggered), and so these rules are also causes of the selected action (again according to a probability relation).

As such, the causality of a given action in *Prom Week* is distributed over potentially thousands of social concerns. Moreover, this distribution may be fairly even: it may be the case that the social concern that contributed most to the selected action being taken is one that accounted for an extremely small percentage of the total utility computed for that action. This is how causality works in the real world—most relevantly here, the causality driven by social norms—but it is not how it works in narrative, where contingency is most important. Just as characters are exaggerations and simplifications of human beings, causality in narrative exaggerates and simplifies the corresponding real phenomenon.

In the case of *Prom Week*, this extreme richness supports their design goals more than it hinders them, since the aim was to produce a game that would support emergent puzzle solutions according to a kind of *social physics* [799, p. 235]. Indeed, as Alex Mitchell has noted, through gameplay this underlying richness can be apprehended by the player:

> On repeat readings, the surface behaviour I observed began to reveal

[65]Here, I am simplifying in a particular way, since characters do not autonomously decide which action to take, but instead determine the handful of actions they would like to take most, which a player then selects between. However, the recipient of an action selected by the player will autonomously decide how to respond to that action; in *Comme il Faut*, this means either *accepting* or *rejecting* the action [804, p. 100].

> the underlying system, and I began to develop an understanding of this system through interaction. [841, p. 34]

As Mitchell explains, this kind of understanding of an underlying system through surface interaction is what Noah Wardrip-Fruin has called the *SimCity effect* [1311].[66] But this understanding must come in the form of a general sense of the simulative mechanisms, since no human could fully comprehend the presence and interplay of several thousand social rules. Indeed, the basis of such understanding must be a capacity to aggregate the fine-grained effects of many rules into a coarser representation of essential contributing factors. Could such reasoning be carried out automatically, as a way of binding character actions to construct an emergent contingency structure?[67]

I was not a member of the *Prom Week* development team, but in 2015 I worked on a collaboration that sought to automatically generate narrative summaries of gameplay that would be structured as emergent contingency structures [44]. Specifically, we planned for such a generated summary to appear at the end of a playthrough as a diary entry written from the perspective of a character who had been central during gameplay. It would be a generated story recounting what had happened during the playthrough. To enable such procedural generation, we needed an automatic means for identifying why characters took the actions they took and how those actions related to one another in terms of contingency.

Our general approach to this challenge was to isolate a small set of social rules whose execution led to a given character action being taken. The identification of

[66] In a recent email exchange, *Prom Week* cocreator Mike Treanor noted that the game's interface does much of the work in enabling this kind of player understanding, primarily through explicit surfacing of some of the hidden mechanisms of the underlying simulation (personal communication, April 23, 2018).

[67] The inferred action relations would not quite be contingency ones, but rather fine-grained probability relations aggregated into coarser-grained relations that could potentially be understood as *quasicontingencies*. In any event, inferring such relations would yield a more actionable comprehension of the causal forces at work.

such a set, we figured, would hinge on a particular numeric value: the difference between the utility computed for the selected action and the utility computed for the runner-up candidate action.[68] Let us call this utility difference x. So, for example, if at some decision point the utilities of the two highest-scoring candidate actions were computed at `122` and `81`, respectively, then x would be `41`. Having determined x, a system could then identify a set of executed social rules whose utility valences summed to at least x. This set of executed rules would be meaningful because, had they not all fired, the other candidate action would have been selected instead. Moreover, such a set of rules could be utilized to automatically generate a causal explanation for the action (one that would express the social considerations encoded in the preconditions of those rules).[69] This procedure would be an example of *abductive reasoning* [25], where explanations are generated to account for observations.

While this seemed like a good approach, it did not work in practice. The problem was that such a rule set might comprise several dozen rules, but that would make the resulting causal explanation too rich for human comprehensibility. Next, we considered isolating only the top three contributing rules, but in many cases these rules, even when taken together, could not produce a compelling causal explanation for an action (since they may have only captured a small per-

[68] Again, matters are confused by the fact that some character actions are selected by players from a pool of high-scoring candidates. It is not worth dealing with this subtlety here.

[69] Preconditions operating over the *social facts database* [800, p. 5] would be especially useful, since entries in that database represent past events. This could allow for explanations like 'Zack decided to ask out Monica because Monica had done something funny recently', or more complex examples such as 'Doug insulted Oswald because Oswald recently bullied Monica and Doug considers Monica a friend'. We also considered the prospect of including negative utility valances in this process. For this, we could consider a set of negative utility valances for the second highest-scoring action whose absolute values sum to at least x. A set of rules retrieved in this way would capture that the selected action would not have been taken if this set of fired rules had not decreased the utility of the runner-up (to make it less appealing than the selected action); this is an example of *counterfactual reasoning* [905]. A hybrid approach is also possible, meaning one that combines both rules that added to the utility of the selected action and rules that subtracted from the utility of the runner-up.

centage of the total utility computed for that action). In the end, we more or less abandoned our goal of reasoning over utilities to infer causality, because the utility computation in *Prom Week* was too rich to provide causal accounts that would be comprehensible to humans. Since it did not seem possible to discern the cause of even a single action, the prospect of inferring something on the order of an emergent contingency structure was out of reach. We could not tell stories about *Prom Week* gameplay.

Contingent Unlocking in Klein's System

How can a simulation facilitate the kind of reasoning that is needed to discern emergent contingency structures? It turns out that Klein's system, as early as it was developed, provides an elegant solution.

One special kind of action effect in this system that I did not discuss in Section 4.1.3 triggers updates to a critical set of *story registers*. These are data stores that record information about emergent events as a way to guide the generation of subsequent emergent events. For example, during the daytime, all the characters engage in various social activities at Lady Buxley's countryside manor—tennis, chess, conversation, drinking together—and over the course of such an activity two characters may flirt with one another (depending on their personalities, affinities, and other factors). When two characters emergently flirt while being observed by another character, the system records this information by adding a `(flirter, flirter, observer)` triple to a special flirtation story register.

Later on, either during the daytime activities or at night after everyone has gone to bed, a special follow-up event is unlocked: the flirters may run off together for a secret tryst that is witnessed by the original observer. Once this happens, a suite of additional emergent follow-ups may be unlocked, depending on the relationship between the observer and the lovers. For example, if the observer's spouse is among the lovers, the observer may acquire a motive for killing either the spouse or the other person. Alternatively, the observer may threaten one or both of them for the same reason, in which case the threatened parties may obtain a motive for killing the observer. In the case that a spouse is not involved in the tryst, the observer may blackmail the other parties, who will thereby obtain a motive for killing the observer. Just as a flirtation register tracks the characters who qualify for having a tryst, a motive register compiles all the characters who have unlocked the affordance of murdering another character. Finally, the murder scene occurs emergently when a character from the motive register attacks one of the characters for whom there is a motive to kill.

Thus, in this way, emergent actions in Klein's system *unlock* future emergent actions that may build upon them to grow full-fledged emergent storylines—such unlocking is an expression of the contingency relationship (causality with a simple and legible cause), and thus such emergent storylines are precisely what I have termed 'emergent contingency structures'. I call this method *contingent unlocking.* For me, the phenomenon suggests the image of a plant growing out of a spongy forest bed: meaning is constituted against the soft backdrop of entropy.[70]

Causal Bookkeeping

While the kind of causality at play in *Prom Week* is a latent one—actions change the world, and the state of the world determines subsequent actions—causality in Klein's system is *explicitly* modeled by the system in terms of contingency relations.[71] Beautifully, this allows the system itself to also *identify*

[70]In an email discussing prospects for collaboration, Angus Forbes has proposed visualizing emergent narrative using this explicit representational metaphor (personal communication, November 5, 2017). I think that is an exciting idea.

[71]To be clear, there are visualizations of *Prom Week*'s playtrace data [804, p. 108] that resemble the graphical structure of Klein's system's narrative possibility space as depicted in

the explicit causal relations that bind actions. Thus, by letting emergent actions explicitly unlock subsequent emergent actions—again, I call this 'contingent unlocking'—emergent contingency structures can not only be generated, but also be easily identified later on.[72] To facilitate such identification, however, the system should explicitly record contingency relationships between events as they occur—I call this technique *causal bookkeeping*. As I discuss in Part II of this dissertation, contingent unlocking and causal bookkeeping have become the most important tools in my technical practice.[73] Unfortunately, Klein's generated prose does a poor job of rendering the emergent contingency structures that are enabled by its contingent unlocking, in part because it does not do causal bookkeeping in tandem: while emergent events contingently unlock subsequent ones, the system does not record this information for exploitation at the level of curation. Later, in Section 4.2.3, I discuss how doing so would have allowed the system to produce far better narration of its murder mysteries.

So, the takeaway for this section, and perhaps this entire chapter, is that simulations that do contingent unlocking and causal bookkeeping will be far more likely to show not only a capacity for generating satisfying emergent storylines, but for identifying (and ultimately surfacing) them as well. As such, these techniques may be thought of as underpinning a regimen for good *causal hygiene* in character simulation. Like many of the ideas reported in this dissertation, my thinking on this subject was first presented in a 2015 paper called "Open Design Challenges

Figure 4.3, but these are different in kind: rather than causal event sequences, these illustrations of playtrace data represent *temporal* sequences (actions that players took).

[72]The task of identifying such sequences in the larger morass of simulated material is *story sifting*, and it forms the backbone of my framework for curationist emergent narrative. Story sifting is explained in Section 5.3.1.

[73]After five years of thinking and coding, I independently converged on a similar approach to Klein's. It was not until I read Marie-Laure Ryan's lucid account of his system [1065] that I realized he had essentially already solved the problem. I think this kind of technique is one that is so elegant and clearly well-founded as to seem obvious once it has been noticed.

for Interactive Emergent Narrative" [1058].

Contingent Unlocking in ***Prom Week***

Briefly, before concluding this section, I would like to note that *Prom Week* actually *does* utilize contingent unlocking, but it was introduced toward the end of the game's development and does not feature heavily in the final product (or its scholarly reporting). This mechanism comes in the form of *story sequences*, which are emergent sequences that play out in a series of explicitly connected *instantiations*. In *Prom Week*, once the player has selected an action for a character to take with another character, the system decides how the receiving character will respond, before finally rendering a concrete performance of the action by the two characters [803, p. 4]. These realizations, or 'instantiations', couple generated natural language and character animation. Story sequences, then, emerge at the level of a series of instantiations, where explicit and sometimes evocative connections are made to connect the individual instantiations. For example, one story sequence follows the arc of *A Christmas Carol* [262], with instantiations concerning visits from Jacob Marley, the Ghost of Christmas Past, the Ghost of Christmas Present, and the Ghost of Christmas Yet to Come, as well as a conclusion scene in which a *Prom Week* character realizes the error of her ways.[74]

I do not believe the notion of story sequences has been discussed in any of the papers on *Prom Week* or its technical substrate, but a few years back cocreator Ben Samuel gave me an overview via email:

[74]Personal communication with Ben Samuel, April 30, 2018. Though for its developers it was a beloved element of the game, *Prom Week* lead author Aaron Reed once told me that he was not sure that any player had ever actually experienced this remarkable story sequence. This was because it was authored in such a way that it could only emerge out of very specific gameplay contexts, which suggests a pitfall that is germane to all computational media: authoring effort may be exerted to produce content that is rarely or never encountered in experiences with a work. Indeed, Ben Samuel recently informed me that he authored no less than fifteen instantiations for this particular story sequence (personal communication, April 30, 2018).

> Story Sequences was an idea that we got really excited about, but either through our execution, our presentation, or a combination of the two (or something else entirely), they were never a particularly well advertised part of the system. A story sequence, essentially, was a mini-story arc; a story that spanned multiple instantiations. [...] But how could you do this in Prom Week? With all of the different instantiations, how could you guarantee that the first instantiation in the sequence came before the second? Why, with [labels in the social facts database], of course![75]

In *Prom Week*, a special data store called the *social facts database* (SFDB) files records that index simulated events according to a set of author-defined social labels [804, p. 101]. For instance, if one character insults another, the system might file the insult under the label `rude`.

To support story sequences, the authors invented a hack of this labeling mechanism by which the component actions of a sequence are filed under special sequence-specific labels (e.g., `SFDBLabel_SS8_1` for the first action in the eighth authored story sequence). In turn, later actions in the sequence will have preconditions that demand the presence of such sequence labels in the social facts database (so that, for example, the second instantiation in a story sequence does not appear out of context). Moreover, to increase the likelihood of story sequences emerging, the game's creators authored rules that would boost the computed utility of the components of story sequences.

It should be clear by now that this SFDB hack bears a striking resemblance to Klein's story registers. While the system does not explicitly do causal bookkeeping, it could be trivially implemented by recording cases of the special SFDB labels unlocking the next instantiation in the story sequence. Unfortunately, as Samuel noted in his email to me, *Prom Week*'s story sequences did not end up becoming a major part of the gameplay experience or the scholarly reporting on

[75] Personal communication, March 8, 2014. My explanation of story sequences relies on the information provided in this email.

the system. Nonetheless, the subsystem is a nice example of contingent unlocking in emergent narrative.

So while Klein's murder-mystery generator and *Prom Week* both make use of contingent unlocking to support emergent contingency structures—what I believe to be the holy grail of emergent narrative—no one seems to have noticed. As with many of the other pitfalls outlined so far, this appears to be due to the relatable case of developers who could not quite close the loop on execution (or reporting) due to a lack of time, energy, or other resources. Moreover, neither system explicitly carries out causal bookkeeping, but in either case it would have been feasible, as I have shown. Nonetheless, each project's approach to contingent unlocking serves as a good example of the kind of technical approaches that can be utilized by practitioners building their own systems.

4.2 Curation Pains

While simulation issues may hamper the quality of stories produced in works of emergent narrative, curation issues trouble the very classification of those artifacts as stories. In this section, I begin with what I have identified to be the essential flaw of emergent narrative—the mere transpiring of simulation is conflated with narrative—before proceeding to articulate a series of additional curation issues that have worked to hamper the form over the course of its entire history.

4.2.1 No Telling

There is a troublesome conceit in emergent narrative that the mere transpiring of a simulation, or of an interaction with one, constitutes narrative. This notion is especially prevalent in interactive works, including games such as *The Sims* [792] and academic projects like *FearNot!* [55]. My view, as I alluded to above and explain more thoroughly below, is that a narrative artifact only manifests through actual curation of the simulation stream. That is: *raw simulation is the stuff of emergent narrative, not the shape of emergent narrative.* I am not the first person to offer this kind of critique, and in this section I will make heavy use of specific arguments by earlier writers.

The Raw Unfolding of Time is Not a Story

As Espen Aarseth notes in his seminal *Cybertext* (1997), the idea that simulation is a kind of narrative is likely rooted to some degree in the *narrative turn* [616, 233] (or narrative turn*s* [521]) in the humanities and social sciences:

> There is a tendency in much cultural theory to posit narrative as the grand structure of everything, the foundation upon which we order our lives and actions. To suggest that narrative is not wholly deserving of this reverence might be risky, since it is all too easy to point out that even the very point I am making here could not be made without the support of narrative. [6, p. 94][76]

I too am reliant on a kind narrative mode in this very writing, and I am in fact someone who subscribes to the ideas that drive the narrative turn, but it is critical to realize a subtle distinction: narrativist scholars and researchers do not posit narrative as the basis of human experience, but rather as the basis of *human sensemaking.* When narrative psychologists write of the "storied nature of human conduct" [1093], they do not mean that human conduct itself constitutes narrative—that the raw unfolding of time is literally a story—but rather that *we can only understand it as a kind story*, since we process the world and our

[76]Later, building off of Markku Eskelinen [317], Aarseth would also identify a "narrativistic colonialism" by which "the (academic) discovery of computer games over the last two decades is accompanied by the most smothering form of generic criticism: the attempt to reform games into a more acceptable form of art, literature or film; i.e., as narratives" [4, n.p.]. Scholars such as Jesper Juul backed this view [557, p. 16], while others took offense at the phrasing [869, 1161].

experiences in it through a mechanism of narrative sensemaking. As Theodore R. Sarbin makes clear [1094], narrative by this perspective is a *root metaphor* for human conduct. The notion of a root metaphor is due to the philosopher Stephen C. Pepper [930], who defines it as a fundamental organizing principal for sensemaking (both formal and informal):

> By a root metaphor, I mean an area of empirical observation which is the point of origin for a world hypothesis. When anyone has a problem before him and is at a loss how to handle it, he looks about in his available experience for some analogy that might suggest a solution. This suggestive analogy gives rise to an hypothesis which he can apply towards the solution. The method of development of world hypotheses for the problem of gaining comprehension of our world, follows, I find, the same procedure. [929, p. 3]

Thus, narrative by this view is a mechanism that can be applied to make sense of the world, *and that application works metaphorically*, since the raw material of the world is treated as narrative *by metaphor*. But, again, this does not mean that the raw material is literally narrative.

While narrative psychologists (and humanists and social scientists and philosophers and other professional thinkers) have applied narrative sensemaking to make sense of their subject phenomena, they also describe how all human beings carry out this same procedure to make sense of everyday experience. In this way, as the basic mechanism of human understanding, narrative sensemaking drives conception of all human conduct, and moreover conception of all conduct whatsoever, inasmuch as human understanding obtains. In fact, this sensemaking is actually *overactive*—studies have found that humans may even perceive random event sequences as constituting narrative [1094].[77] Still, while essentially any happening

[77]Richard Walsh provides an interesting temporally oriented account of the human tendency to narrate: "Narrative, after all, is our principal way of understanding our experience of reality by articulating patterns in time; if this cognitive strategy misleads us in relation to emergent phenomena, it may be something to do with the fact that narrative itself—as instance, as mode, and as cognitive faculty—is itself an emergent pattern in time" [1308, p. 84].

may be understood as constituting a kind of narrative (or at least eliciting narration), this does mean that the happening is literally a story. To maintain as much is to conflate metaphor with an equivalence relation.[78]

The Raw Unfolding of Simulation is Not a Story

Now, we may return to Espen Aarseth, who is likely the scholar most associated with the argument that raw simulation is not literally narrative. On the same page in *Cybertext*, Aarseth continues:

> But the story of an event is not necessarily the same as the event itself, and stories can be told about things other than stories, luckily. [...] For instance, both stories and games of football consist of a succession of events. But even though stories might be told about it, a football match is not in itself a story. The actions within the game are not narrative actions. [6, p. 94][79]

In his own invocation of this sports analogy, the scholar Markku Eskelinen did not mince words:

> if you actually know your narrative theory [...] you won't argue that games are (interactive or procedural) narratives or anything even remotely similar. Luckily, outside theory, people are usually excellent at distinguishing between narrative situations and gaming situations: if

[78]If all phenomena are understood through narrative sensemaking, then stories themselves must also be processed in this way, which means that humans mentally construct (meta)stories to make sense of stories. Does this then mean that there are no stories constituted in the external world, but instead only stories that humans tell themselves about those stories? While the subjective nature of story comprehension is likely unavoidable [68, 276, 407], this line of thinking is probably not that constructive here. By assassinating objectivity, as we did in Section 3.1.1, we can destroy all delineation and demarcation, but then we are not left with much to talk about. Just as there is a thing called 'nonfiction', there are things called 'stories', and so for the purposes of this dissertation I will assume that 'stories' are in fact stories. (A thought: if humans must construct metastories to make sense of stories, then it follows that in turn humans must construct metametastories to make sense of the metastories, and so forth to some limit of recursion rooted in either human understanding, human metaunderstanding, brain physiology, or the material universe.)

[79]Incidentally, as noted in Section 2.2, in her first articulation of the approach, Ruth Aylett explains emergent narrative by making an analogy to the ways in which "recognizable narrative structure" emerges in a soccer match [50, p. 84].

> I throw a ball at you, I don't expect you to drop it and wait until it starts telling stories. [317, pp. 175–176]

Later, in a 2004 paper, Aarseth argues that stories made up of simulated events are typically constructed *after the fact*, by the process that I call curation:

> Nevertheless, games like *The Sims* are sometimes (not often) used as storytelling machines, when particularly memorable moments in the game are retold by the player/god. But this is not translation from game to story, this is simply good old after-the-fact narration, like the football column in the Monday sports section, the lab experiment report, or the slide show of one's Carribean vacation. Something interesting happened, and we want to tell others about it. Ontologically, the capacity for generating memorable moments is something games have in common with real life, as well as with stories. A story-generating system does not have to be a story itself. In fact, while life and games are primary, real-time phenomena, consisting of real or virtual events, stories are secondary phenomena, a revision of the primary event, or a revision of a revision, etc. [4, n.p.]

In the interim between these papers—and in the midst of the *ludology vs. narratology debate* in game studies [356, 869, 767], which also structures Aarseth's post-*Cybertext* writing—Jesper Juul argues that there will tend to be (or there necessarily will be) fundamental divergence between gameplay as experienced and gameplay as recounted:

> if we recount a game of chess, our playing of the entire Half-Life game or a multi player game of Starcraft, the existents and events will be transferred, but not the dynamic systems. Our retelling will not be a game, and in fact much of the vast journey that it takes to complete Half-life would be excruciatingly dull if retold in any detail. [555, n.p]

Finally, in a more recent 2012 paper, Aarseth pains himself one last time (for now) to consider the view under fire in this section, which he terms *narrativism*:

> This conflation of any kind of diegetic or experienced situation with storytelling is what I have previously labeled narrativism [...] If an(y) interesting experience in a game is an "emergent narrative," where

> does it end? And why limit this category to game-based situations? At some point it becomes hard to distinguish narratives from any other type of worldly experience, at which time (or long before) we might as well give up the discussion. [5, p. 131][80]

While Aarseth and his peers are specifically concerned with videogames and interactivity, I think they hold just as well for systems outside videogames (such as story generators) and for noninteractive cases. Indeed, the narrative theorist Richard Walsh argues that, in any case, emergent narrative obtains because we are inclined to narrativize (alternatively: curate):

> a particular run through a simulation [...] does not in itself produce emergent narrative but instead emergent behavior. So Aylett's virtual Teletubbies, interacting with each other and with the user's input will (given the right conditions) produce emergent behavior that we as observers may be inclined to narrativize. Similarly, the goal-oriented user in [an] interactive environment participates in the emergent behavior that system provides for, and that *experience* is one that the user may be inclined to narrativize. [1308, p. 77]

Indeed, this misconception seems to be fundamentally rooted in the potential *tellability* of simulated events. As I explained in Section 4.1.1, in the study of personal narrative, events are deemed tellable if they would make for a good story (or a story at all) [882]. It would be strange, however, to argue that the tellable events of lived experience literally constitute a story—the story, of course, materializes in the actual telling. But this idea is what drives the troubling argument that the mere transpiring of simulated events constitutes a story. According to this perspective, a simulated event, *as it transpires*, constitutes a segment in an

[80] In his account of the ludology vs. narratology debate, Frasca uses the term 'narrativism' to refer to the stance taken by the contingent conventionally called the 'narratologists' [356]. (He does this, following Michael Mateas [764, p. 32], from the belief that 'narratology' was being used incorrectly in the debate [356, p. 2].) This does not appear to be the sense with which Aarseth uses the term. Instead, as he has noted [4, n.p.], he is invoking what Alan Rauch has termed "story fetishism" (extremism in the narrative turn outlined above). Curiously, my attempts to track Rauch's usage of that phrase only turn up Aarseth's quotation.

unfolding story. By analogy, a tellable event in the real world, *as it transpires*, would constitute a segment in an unfolding personal narrative. This formulation is strange because it mixes up what is being recounted with the actual recounting. The unfolding of time itself is not a story, and so neither then is the raw unfolding of a simulation (or some record of that unfolding, such as a simulation trace). Put another way: simulated material is the *stuff* that composes emergent narrative, but it is not itself emergent narrative. To contend otherwise is to conflate simulation with narration.

But Simulation Does Afford Narration

There is still something powerful happening here. While tellable events are not a telling, tellability critically *affords* telling—in turn, while simulated events are not stories, they may afford stories. Thus, the fact that simulated events may be tellable at all has a major implication: stories may be told *about* simulations, which means the very *project* of emergent narrative is viable. This project, however, is predicated on a procedure of curation.

Let us briefly recall Hayden White's model of historiography, which I outlined in Section 3.1.1. White argues that an historical account may only materialize through an act of curation, by which the raw material of transpired phenomena (White's 'historical field') is processed and then embedded into the structural framework of a story. This process entails several steps: the phenomena transpire, a chronicle (lossily) records them, a subset of the chronicle is selected to be included in a story, and finally the story is fit into a particular 'emplotment' (e.g., the tragedy or comedy), resulting in an historical account with targeted rhetorical features. Just as historical accounts obtain through curation of an historical field, emergent narrative obtains through curation of a simulation.[81] By such a

[81] Jan Simons has also noticed a common essence spanning historiography, personal narrative,

procedure, tellable simulated events may actually be told.

Offloading Curation to Interactors

Finally, we arrive at the fundamental flaw of emergent narrative: there is no telling. By implying that their raw simulation streams literally constitute narrative, works in this area offload the burden of curation—by which stories may actually obtain—to human interactors. In this way, they are story generators that do not actually tell stories.[82]

To be clear, I think this pattern can work quite well in some cases—*Dwarf Fortress* and *The Sims* cannot be denied—but it requires a good simulation and a human willingness to curate. This human curation can take the form of basic narrative sensemaking (players mentally narrativizing the unfolding simulation), but more powerfully it may result in conventional narrative artifacts that are mounted in full-fledged media experiences. Here, Tim Denee's comic *Oilfurnace* [255], the subject of my case study of Section 3.1.1, is a perfect example. Another is Robin Burkinshaw's *Alice and Kev* [150, 325], which was produced by curating *The Sims 3* [1207] gameplay. In his recent dissertation, Eric Murnane provides an extensive analysis of player stories that recount emergent experiences in *The Elder Scrolls V: Skyrim* [867], a practice that he himself has carried out with the extensive 'narrative photojournal' *Three Hundred Days in the North* [866]. Like-

fiction, and emergent narrative: "events, whether historical or fictional, actual or virtual, are not intrinsically narrative or non-narrative but they become stories because someone deems them 'tellable' and perceives or construes them as causally and chronologically connected" [1161, n.p.].

[82]Furthermore, a loose idea of emergent narrative being constituted in simulation may lead to a vague conception of what form stories actually take in a project. Hartmut Koenitz has raised this concern with regard to a project that he views as otherwise being successful (citation inserted for clarity): "From this perspective, McCoy et al.'s otherwise outstanding work on Prom Week is marred by the lack of a clear definition of 'story.' For example, in one of their papers [803], the authors variously refer to story as 'personally meaningful' game experience, as 'campaign,' as 'gameplay,' as a 'collection of levels [...] where the player can take social actions,' or they talk about 'player agency at the story level' and the 'player's path through the story.' Each of these descriptions depicts 'story' as a slightly different concept" [611, p. 10].

wise, Stephanie Boluk and Patrick LeMieux write about the 'Dwarven Epitaphs' produced by players of *Dwarf Fortress*, among which *Oilfurnace* is a featured example [126]. While I view these as powerful examples of curationist emergent narrative, the nucleus of this dissertation, this pattern of human curation does not seem to be prevalent outside the realm of popular videogames. In the case of non-interactive story generation, and indeed many other interactive systems, humans may not be so inclined to tame the welter. To again invoke my coadvisor Michael Mateas's critique: without curation, emergent narrative is *just one damn thing after another*.

Toward Automatic Curation

What I seek to promote in this dissertation is not just an emphasis on the importance of curation in emergent narrative, but moreover an approach *by which systems themselves may actually tell stories*. This is no small task, since it would entail the *automatic curation* of a simulation in a manner that is (in terms of procedure) akin to Hayden White's historiographic method: the system must sift through a record of what has happened in order to identify interesting material out of which a narrative artifact may be constructed. That is, the system must not only be able to tell stories, but it must also have the capacity to discern tellable event sequences in the first place. Further, as I will argue in Section 4.2.4, curation should not culminate merely in a narrative artifact, but moreover that artifact should be mounted in a full-fledged media experience. In the next chapter, I fully articulate my vision for procedural narrative in the form of my framework for *curationist emergent narrative*.[83]

[83]The essence of my argument here first appeared in two 2015 papers [44, 1058] that introduced a challenge that I refer to as *story sifting* (then called 'story recognition'): the task of automatically recognizing interesting narrative material embedded in the morass of accumulated simulated material. The latter paper, in particular, may be viewed as the embryo of this

4.2.2 No Curation

While works of interactive emergent narrative typically do no telling, simulationist story generators may do telling without curation. Whereas in the former case the emergent narrative is viewed as obtaining through the process of a simulation transpiring over time—with its events washing over the interactor, if you will—in the latter case the evolution of the simulation over time is captured in a *trace*, and that trace is what is (mis)treated as a story.

***Tale-Spin*'s Curious Model of Creative Writing**

This approach is probably best characterized by early systems like Sheldon Klein's murder-mystery generator [601, 597] and especially *Tale-Spin* [822]. As Masoud Yazdani writes (in critique, not defense), creator Jim Meehan's approach to the latter project instantiates a peculiar model of creative writing:

> A story [by Meehan's approach] is an account of goal directed behavior of a set of characters—their interactions with each other and with the world. Story writing is simulating a world in which characters follow a goal directed life, presenting the trace of the simulation to the readers. The readers can be expected to be interested in these stories as they are concerned with problem solving and purposeful interaction themselves. They like to find out about such situations and find these stories interesting. [1366, p. 138]

To be clear, Yazdani's account is actually a recapitulation of an explicit claim that Meehan himself proffers in his dissertation:

> What happens when the simulator runs may or may not be interesting [...] How do you make it interesting? You fix it in advance. You rig the world so that if people do behave rationally, they'll do some interesting things. So sitting on top of the simulator, if you like, is a program which knows both about stories and about the world model. It models a writer who has something in mind that he wants to tell a story about. [822, p. 108]

dissertation. In Section 5.5, I recount this intellectual development in more detail.

Indeed, the *Tale-Spin* approach may be criticized for its idea of how stories are made—this is what sparked the turn, beginning with Natalie Dehn's *Author* [252] and Yazdani's *Roald* [1363], toward an author-centric approach to story generation that more intuitively models the process of human story invention. For our purposes, however, I want to call into question a specific feature of this approach, which is its idea of *how a story looks*.

A Simulation Trace Is Not a Story

Tale-Spin's generated tales are essentially prose renditions of traces through its simulation, and I believe this aspect of the project has contributed to its lambasting by humanists and later practitioners. Here is an example excerpt from a system output:

> Tom decided that Wilma might want Tom to give Wilma a worm. Tom wanted to ask Wilma whether Wilma would tell Tom where there were some berries if Tom gave Wilma a worm. Tom wanted to get near Wilma. Tom walked from the chair across a living room down a hall via some stairs down a hall down a hall through a volley across a meadow to the ground by the redwood tree. Tom asked Wilma whether Wilma would tell Tom where there were some berries if Tom gave Wilma a worm. [822, p. 232]

A trace of an entire simulation over time does not make for a story, but rather a recording of a *storyworld's history*. In the case of an interactive experience, an interactor may be willing to process the welter of ongoing simulation to sift out the emergent stories, but this is probably not to be expected in the case of a prose artifact. You likely struggled to finish reading even the short excerpt that I just quoted. In Section 4.2.4, I discuss how presentation issues have contributed to much of the failing of emergent narrative, but presentation only comes at the end of a substantial process of curation. Before Meehan's *Mumble* system generates its stilted prose, the material that will be presented has already been selected for rendering in natural language. The process of deciding what material should be included in the story presentation is an early step in the curation procedure, and one that *Tale-Spin* fails to adequately carry out. By the time that its generated prose does further injustice, the system's tales are already doomed because nearly all of the available simulated material is blindly selected for telling, which makes the outputs more like simulation traces than stories.

Simulation Traces as Whitean Chronicles

We can think of the curation procedure of emergent narrative as working like Hayden White's historiographic procedure (discussed above): an historical field (simulation) is captured in a chronicle (trace or other data), out of which a subset of material is selected for use in a story (story) that is embedded into an emplotment (is deliberately crafted) with particular rhetorical features.

When a monolithic simulation trace is printed out, a Whitean *chronicle* is produced, not a *story*. Indeed, Stephanie Boluk and Patrick LeMieux have made essentially this observation with regard to the simulation traces found in *Dwarf Fortress*'s Legends mode [279]:

> *Dwarf Fortress*'s linguistic forms of historical inscription in Legends mode bear a striking formal and thematic resemblance to early forms of writing, such as the medieval annal and chronicle. [126, p. 144]

As I discussed at length in Section 3.1.1, White explains how chronicles have no beginning (they merely start), no end (they merely terminate), and no meaningful structure in between. To tell a story out of external material—this is the essential task of historiography—the teller must carry out a process of deliberate selection and deliberate assembly of the selected material into the frame structure of narrative. To be clear, this means selecting a subset of the material contained in the chronicle. If instead a chronicle is itself presented as a story, the recipient will deem it either dubious or non-narrative—she will likely be unwilling to sift through it to find the story (only historians and lawyers do this). This, I contend, is what drives the harsh critiques of Klein's murder mysteries and *Tale-Spin*'s fables—simulation traces are chronicles, not stories.[84] Later, in Chapter 8, I apply this critique to my own work in a takedown of my project *Diol/Diel/Dial*.

Klein on Curation

To be clear, earlier practitioners did realize the importance of curation to some degree. Meehan knew that his simulation traces were not good stories, and Klein admitted, "It is a rather long distance from the promise of the title [a novel] to the fragmentary output just presented" [601, p. 30]. Later, in an interview with *Byte Magazine*, Klein explicitly emphasized the importance of curation with regard to his computer-generated video opera *Revolt in Flatland* [602]:

> Klein and his colleagues showed a video-taped sample at the Fifth International Conference on Computers and the humanities, in Ann Arbor, Michigan, last year. But, says Klein, "I'm not keen on showing it because, to be perfectly frank, the music is superb but the action is quite dull." This is because the action simulator treats the cast's movements as if they were taking place on a realistic scale, rather than compressed onto a stage. "Most of what's happening in the current version is that little squares and triangles and polygons are moving slowly from one house to another. The action only happens

[84]Boluk and LeMieux, who also connect simulation traces to Hayden White, do so with favorable appraisal of the former: "By contrast, White argues that the annal and the chronicle are not ideologically primitive forms of inscription but instead embody a fully developed but radically different philosophical worldview. He proposes that these forms of medieval writing have been misrecognized in much the same way that it is easy to misinterpret the output of *Dwarf Fortress*. Like *Dwarf Fortress*, the annal deprivileges the place of the human within a larger cosmological landscape" [126, p. 145]. This line of argument gibes with my call, coming in Section 3.1.4, to embrace a new aesthetics of the computational—a kind of *computer art brut*—but I am not sure simulation traces are the right artifact for the job. Often traces seem to be the lazy first approximation of an actual considered artifact. I think we can do full-fledged emergent narrative without losing the evocative aesthetics of the computational, but doing so means carrying out a procedure of curation. This is the primary call of this thesis.

> occasionally, when they meet," Klein says. A refined version of *Revolt in Flatland* is being prepared which automatically omit the boring scenes. In the meantime, says Klein, "Potentially, our opera could be five hours long, like a real opera, but at the end there would be nobody watching." [817, p. 51][85]

Saga II's Brilliant Curation

Moreover, one of the early projects actually demonstrates an emphasis on curation that would be exceptional even today. In *Saga II* (1960), discussed in Section 4.1.2, there is no conceit that raw simulation traces are narrative artifacts or full-fledged media experiences. Instead, the printouts containing such traces were subjected to additional curation, by which shooting scripts were developed and further curated to drive the production of a television scene. This emphasis on curation—and mounting in a full-fledged media experience, which is the call of Section 4.2.5—was likely due to the project being a collaboration between computer scientists and television producers. Nonetheless, it makes *Saga II* a fabulous example of the power of curation in emergent narrative. In Chapter 5, I celebrate this achievement further.

Even Traces Are Curated

Before moving on, it is worth noting that even chronicles and interactive simulations result from a process of curation. Indeed, as I will explain in Section 4.2.6, *Tale-Spin*'s simulation traces are in fact highly curated with regard to the larger

[85] Unfortunately, a refined version was never produced. I reached out to Mark Manasse, one of several students who worked on the project with Klein (it was being undertaken for his seminar class), to ask what happened. As he explained, a strike critically halted work on the project (and presumably blocked computer access, as well): "We had a lot of ambition, but the class coincided with a strike by the teaching assistant's union, which greatly interfered with [our] implementation plans [...] to allow our ambitious system to do its work; the design existed on paper, but never got finished" (personal communication, April 20, 2018).

accumulation of material that is produced in a given run of the system.[86] The same goes for Sheldon Klein's murder-mystery generator. This is actually quite intuitive: when a simulationist method is employed, the system is not operating at the level of natural language, or even plot. Rather, a world is simulated, and then natural language is generated by a separate procedure to express what has happened in that world. Sheldon Klein and Jim Meehan were very explicit about this distinction, which is why Klein presents his system as primarily a contribution to natural language processing, while Meehan apologizes for his hasty *Mumble* hack that was cobbled together in a day.

Representation is Curation

What I am getting at here is that a simulation cannot be presented, or even experienced, in its canonical form. A simulation trace is a *report* about a world, clearly, but even the storyworlds of interactive simulations cannot be truly encountered—humans cannot actually experience the text worlds of interactive fiction or *multi-user dungeons* [531, 1369, 1067], and this is true also of graphical worlds, be they 2D or 3D. Indeed, the very descriptors 'text world' and 'graphical world' reveal this: those worlds are not actually made of text, and they are not actually made of pixels, but rather that material is the stuff out of which *representations* of those worlds are fabricated. We may be tempted to say that such representations *render* the underlying worlds, but rendering is actually a mode of curation—a textual or graphical representation of a world, even if it is interactive, is a *depiction*, and depictions are not their subjects themselves.

As Michael Mateas has argued, technical strategies for the implementation of computational systems are often conflated with the stories we tell about those systems—what he calls the "inseparable, tightly entangled collection of rhetorical

[86] Preview: *Tale-Spin* actually curates *out* all the interesting material.

and narrative strategies for talking about and thus understanding these computational systems" [765, p. 61]. One such rhetorical strategy is textual or graphical rendering of an underlying world, the canonical form of which would be incomprehensible to a human.[87] To use Espen Aarseth's words for cross-purposes, this conflation seems to be rooted in "a spatiodynamic fallacy where the narrative [world representation] is not perceived as a presentation of a world but rather the world itself" [6, pp. 3–4].[88]

[87]Here, we converge on fundamental issues at the heart of the philosophy of computation, and moreover, of cognition. What is a *Tale-Spin* world? If it grounds out as data stored in memory, a series of `on` and `off` symbols, how do those ground out? Is it in the stored physical charge of the electromechanical system inside the computer? But that stored charge will take a different form on a different computer, even if the world is the same ("the same"), and such divergence may also occur across different runs on the same computer. So maybe the world does actually ground out in a symbolic representation, according to a mapping scheme by which some mediator, be it human or computer, can maintain *isomorphism* for the world across its different physical manifestations. But then we hit *Searle's wall*: "For any program and for any sufficiently complex object, there is some description of the object under which it is implementing the program. Thus for example the wall behind my back is right now implementing the Wordstar program, because there is some pattern of molecule movements that is isomorphic with the formal structure of Wordstar" [1124, pp. 208–209]. While this thought experiment is more obscure than his *Chinese room* [1123], it is far more ramifying: "But if the wall is implementing Wordstar, then if it is a big enough wall it is implementing any program, including any program implemented in the brain" [1124, p. 209]. This is the way toward *panpsychism* (the view that all matter, and each grouping of matter, has a mind [1120]), but James Blackmon comes to the rescue. As he lucidly explains [117], mapping schemes are only *frames of reference*—just as any object may be perceived as traveling at any speed according to the properly concocted frames of reference, any (sufficiently big) material system may be perceived as computing anything if the right mapping scheme is concocted. The problem is that it is hard to concoct such schemes—so, yes, while the completed form of this dissertation, as filed, has existed in my wall all along, I had no means (no mapping scheme) with which to discern it. The monkeys have already typed infinite pages, but finding the ones we want means walking in a Borgesian library. In a sense, however, this is what computation is: we build computer programs as mapping schemes that allow us to discern—in the welter of matter all around us, in which everything ever is always already constituted—the particular phenomena that we would like to encounter. We do not create the images, but rather we capture them as they course through the electromechanical network constituted in the machine. The computer is a weathervane. So while all the *Tale-Spin* worlds, mere images, are already all around us all the time, Meehan's software (or Warren Sack's [1073]) provides the mapping scheme by which we may isolate those images, so that we can encounter them.

[88]Here, I am pitting Aarseth's words against him, since he used them to argue for a world essentialism in nonlinear narrative: while a trope in literary studies suggests that the 'world' of a text may be encountered (and explored) through reading, Aarseth clarifies that this is actually a metaphor. His intended point, however, is that such topology is actually present in nonlinear texts: they present worlds that may truly be encountered. I am taking his argument further, past his point, by contending that it is *still* a metaphor in the case of nonlinear texts, and moreover

All *representational strategies* [781] for rendering storyworlds are constituted in *rhetorical strategies* [766], and together these strategies work to compose larger curation procedures by which underlying storyworlds may be encountered by human beings. One possible mode of curation is textual or graphical representation, and another entails the maintenance of an interface through which a textual or graphical world may receive human inputs. At the heart of this document is a further mode, by which the world is curated so that a narrative artifact is constructed for human encounter (and then mounted into a full-fledged media experience). Of course, these are not incompatible modes, and indeed they may be stacked and recombined in interesting ways. My point is that it does not seem possible to avoid curation—so why not take up the call of this dissertation and be deliberate about the process?

No Emphasis On Curation

You may have noticed that I have just defeated my own thesis for this section—that simulationist story generators like *Tale-Spin* have failed in part due to a lack of curation—but I contend that the point still stands. While Meehan's system does do curation (it is impossible not to), it does so in a haphazard way that reflects a lack of emphasis on this critical aspect of emergent narrative. As such, a more precise title for this section would be 'No Emphasis On Curation'. In the next section, I will identify particular ways that curation can go wrong, even in cases where a system builder emphasizes its importance.

---

even in the case of graphical worlds. While they may seem explorable in a phenomenological sense, these media worlds are just images, like the shadows on Plato's cave wall.

4.2.3 Poor Curation

Since the procedure of curation entails nearly all the same work as conventional narrative construction, a number of pitfalls may befall the (automatic) curator. Generally, it seems that things can go wrong in four ways: poor selection of material, poor arrangement of selected material in the telling, poor presentation of that arranged material, and a failure to mount that presentation in a full-fledged media experience. The preceding section was about an extreme case of poor selection of material (the wholesale selection of a monolithic simulation trace), while this section pertains to more practical considerations of the curation phases in which simulated material is selected and then arranged for recounting. In the remaining sections of this chapter, I discuss the issues of presentation and mounting in an actual media experience.

Klein Kills the Suspense

Again, we find in Sheldon Klein's murder-mystery generator a particularly illustrative example. There is one particular blunder that contributes more than any other to the failings of this system's synthetic tales: the killer is always revealed halfway through the story! As Figure 4.3 illustrates, the murder tends to occur about halfway through each simulation run. But while a delayed revelation of the details of the crime (and particularly the identity of the killer) is the hallmark of the murder-mystery genre, Klein's system divulges these details *as the crime is transpiring*, as this example murder scene shows:[89]

> Lord Edward knew that Lady Jane commited adultry [sic]. Lord Edward was enraged. Edward decided to stab Jane. The day was Sunday.

[89] One could argue that Klein's system actually generates *reverse whodunnits* [1280]. In this mystery mode, the audience knows all the details of a crime, but suspense still obtains through the mystery of how a detective will solve what appears to be a perfect crime. This is hard to argue in the case of Klein's system, however, since there is no rich modeling of detection, as discussed in Section 4.1.5.

> The time was the sunrise. Jane awakened early. Lady Jane decided to go for the walk. Jane got up quietly. Jane thought that Edward was asleep. Jane got dressed. Jane went to the garden. Edward followed Lady Jane. Jane saw Edward. Lord Edward had a long dagger. Edward waved the dagger wildly. Lord Edward stabbed Jane screaming. The knife sank deep. Jane struggled weakly. Jane hit Edward. Lord Edward slashed Jane again. Edward said that Lady Jane betrayed Lord Edward. Jane dying covered with the blood. Lord Edward hid the knife. Edward returned to the bedroom. Lord Edward washed off the blood. [597, p. 109]

Though briefly identified by Ed Kahn [558, p. 5] and then later by Margaret Boden [122, p. 300] and Mark Lee [672, p. 19], Marie-Laure Ryan has probably provided the most extensive articulation of this issue of premature revelation in the generated "mysteries". Here is an excerpt:

> Since the murder is narrated before it is solved the stories lack suspense, the essential ingredient of the genre they are supposed to represent. The program produces the historical sequence of events underlying a detective story, but not a well-formed textual rendition. [1065, p. 515]

Whenever there are interesting simulated events but a misshapen narrative account of them, poor curation is to blame. Thus, Klein's system's deficiencies are not rooted in simulation—though the modeling gap of a lack of detection is another critical blunder, as discussed in Section 4.1.5—so much as in curation. The stuff of a good mystery story is produced by the simulation, but the system does not have the means to tell a good mystery, because it does not know how to arrange a subset of the generated material into the mystery emplotment.[90]

Curation and Tellability

Like all narrative devices, the mystery emplotment's reliance on suspense connects back to the work surrounding *tellability* and *story interestingness* (partic-

[90] Here, I take the liberty of using Hayden White's term 'emplotment', which was discussed at length in Section 3.1.1.

ularly *cognitive interest*), which I discussed in Section 4.1.1. David Elson made this connection in his dissertation:

> the creation of mystery is a device to increase the "tellability" of the story by prompting the receiver to have its own goal—to achieve causal closure in its cognitive model of the story's meaning. As receivers, we are given pieces of a large puzzle, one at a time and in a non-random order, and we must find the most likely (or most satisfying) assemblage of all the facts into a coherent whole. [312, p. 305][91]

Thus, while the murder event is itself inherently tellable—following the discussion in Section 4.1.1, murder is an 'absolute interest' [1103], 'human dramatic situation' [1340, 1341], 'generically important element' [357], and an event of 'situational' [615, 478, 479] and 'instinctive interest' [95]—but the other story events are tellable primarily for being causally related to the murder.

Thus, as the crux of the murder mystery, the murder itself is what William Labov would call the *most reportable event*, as discussed in Section 4.1.1. In describing the act of "narrative pre-construction" that is carried out by storytellers in everyday conversational narrative, Labov informally provides an algorithm for what I call *story sifting* (note that I have already quoted this above):

> Pre-construction begins with this most reportable event and proceeds backwards in time to locate events that are linked causally each to the following one, a recursive process that ends with the location of the unreportable event—one that is not reportable in itself and needs no explanation. [632, p. 37]

[91] A quotation of China Miéville that I recently encountered in Aaron Reed's dissertation [984, p. 98] provides another evocative articulation: "These are novels of potentiality. Quantum narratives. Their power isn't in their final acts, but in the profusion of superpositions before them, the could-bes, what-ifs and never-knows. Until that final chapter, each of those is as real and true as all the others, jostling realities all dreamed up by the crime, none trapped in vulgar facticity" [834, n.p.].

How to Sift Story Sequences

Labov's procedure could be carried out automatically in Klein's system by starting with the murder event and then chaining backward along the causal sequence of events that precipitated the killing. Per the discussion in Section 4.1.6, such a causal sequence will specifically be constituted according to *contingency* relations. As I explained there, narratologist Mark Alan Powell identifies contingency as obtaining between two events when one clearly and explicitly causes the other.

In the case of the story whose excerpt I quoted above, backchaining from the murder (most reportable event) across contingency links would lead back to actions including Lord Edward becoming aware of Lady Jane's affair, Lady Jane having the affair, Lady Jane's flirtations that preceded the affair, and the events that lead to the social practice in which those flirtations occurred (e.g., tennis game), and so forth, finally terminating in an action at the beginning of the simulation. In this way, the system would be able to *sift out* (from the larger accumulation of all simulated events) the emergent contingency structure that constitutes the story's *plot skeleton*. In identifying such sequences, a system may automatically distinguish between what Roland Barthes calls 'cardinal functions' and 'catalyzers' (these are the conventional English translations):

> [narrative] units are not all of the same "importance": some constitute real hinge points of the narrative [while] others merely "fill in" the narrative space separating the hinge functions. Let us call the former *cardinal functions* (or *nuclei*) and the latter, having regard to their complementary nature, *catalyzers*. [79, pp. 247–248]

In translating Barthes, Seymour Chatman instead uses the English terms *kernel* and *satellite*:[92]

[92] I prefer these terms, so I will use them instead of 'cardinal function' and 'catalyzer', the latter of which Chatman argues is a mistranslation [177, p. 54].

> Kernels cannot be deleted without destroying the narrative logic. In the classical narrative text, proper interpretation of events at any given point is a function of the ability to follow these ongoing selections, to see later kernels as consequences of earlier. A minor plot event—a satellite—is not crucial in this sense. It can be deleted without disturbing the logic of the plot, though its omission will, of course, impoverish the narrative aesthetically. [177, pp. 53–54]

In the case of emergent narrative, the satellites of a given kernel sequence could be any action that occurs in the background of the simulation. Because simulations model the storyworld with much higher level of detail than print fiction or other conventional narrative forms—this was the crux of my argument in Section 3.1.1 that emergent narrative works like nonfiction—events may occur that have essentially nothing to do with a given kernel sequence. In this case, such an event should probably not even be considered a satellite of that sequence, but rather an occurrence totally outside its orbit. Moreover, in the kind of massive emergent narrative that I am advocating, where many characters live out entire lives in the world, there is a major possibility of multiple kernel sequences emerging, potentially with some intriguing emergent entangling. Indeed, this prospect of many interrelated storylines emerging is the strongest selling point of the approach. Thus, while Barthes argues that there is a distinct kernel sequence at the core of a given narrative world, as it is reported in a work of fiction, in the case of simulation we might consider kernel sequences to emerge whenever a tellable event emerges. That is, when a tellable event emerges, we may trace is causal lineage (in terms of contingency relations) to excavate a series of kernels that constitute an emergent storyline that is indexed by the tellable event, which in Labov's terminology constitutes the most reportable event in that storyline.[93] While Labov's algorithm implies that a story will always culminate in its most reportable event

[93]Of course, if a more tellable event is encountered through the excavation of the causal structure, then the story may be redefined as being about *that* event.

(since the process of excavation is one of backward chaining), this is of course not a real constraint—indeed, by forward chaining across contingency links, a system may also recognize the ramifications of the most reportable event.

Note that the excavated kernel sequence (a plot skeleton) may actually take the form of a graph, rather than a linear sequence or a tree. This is because the kernel nodes (actions or statives) may have multiple parents (actions or statives that caused them) and multiple children (actions or statives they caused).[94] Marie-Laure Ryan, who has also emphasized this distinction between a plot skeleton and the larger accumulation of narrative material in which such skeletons are embedded, references this graphical structure:

> Within the narrative universe, we can distinguish between a raw historical sequence, consisting of all the physical states and events in their chronological order, and a rationalized sequence, in which information is integrated into a graph according to the rules of concatenation. In this second sequence, each action is linked to a motivating intent, and each intent can be traced back through a causal chain to a physical state or event. [1063, p. 322]

In Ryan's formalism, which is introduced in that paper, nodes in the graph may be linked according to relationships beyond contingency. Indeed, in the case of automatic curation, a system could use whatever semantics it wishes to label the edges according to anything that may be of use to the presentation module that

[94] As I discussed at length in Section 4.1.6, in the real world and in a simulation, the true causal structure may actually take the form of a densely interconnected lattice: since all actions change the world and the state of the world determines whether any action may be taken at any time, each occurring action may be viewed as being caused by each and every earlier action. The creators of *Prom Week*, for example, implicitly describe such a causal structure in their articulation of *social physics*: "In the delicate system of social physics, the smallest social change reverberates and impacts the entire system" [799, p. 319]. However, humans seem to adopt simplified models of causality that resemble graphs, rather than densely interconnected lattices, and certainly this is how causality works in stories, as narratologists like Barthes and Chatman and Powell have shown. Thus, to do story sifting properly, a system must be able to reason about causality such that excavated causal structures are topologically on the order of the causal structures that humans tend to work with when constructing and understanding stories. Following Powell's distinction, this means recognizing contingency links, rather than links that capture the causality relationships of possibility and probability.

will be generating an actual narrative account of the graph. Of course, it would also be reasonable to include satellites (using edges that express as much) and any other material that could be of use to the presentation module.

For such purposes, one might harness the expressive power of the *story intention graph*, David Elson's graph-based story schema that I mentioned in Section 4.1.1. In recent work by my colleagues at UC Santa Cruz, the story intention graph has been used as a representation scheme that enables generative story *re*telling [1008, 716, 712, 713] and richer case-based reasoning for *Minstrel*-style story invention [448, 449, 450].[95] Thus, one specific prospect of automatic curation is to sift through the accumulation of material produced by a simulation to excavate kernel sequences that may be used to automatically build story intention graphs, which could then drive the generation of narrative accounts that are expressed in natural language. This was the ultimate goal of my collaborative work on generating *Prom Week* narrative postmortems [44], though we never got past a proof-of-concept stage in which we resorted to the manual construction of story intention graphs from playtrace data.[96]

Fixing Klein's System

Let us now return to Klein's "mystery" generator. While it has no implemented means with which to excavate kernel sequences, there is a technical solution that is made trivial by the elegant causal feedback loops that drive the system's emergent narrative. As I explained in Section 4.1.6, the unfolding of murder stories is critically enabled by a series of *story registers* that track important information about

[95]The formalism's utility for carrying out automatic reasoning about *story analogy* was established by Elson himself [311]. Inspired by the work of Keith Holyoak and Paul Thagard [501], Elson's approach operationalizes analogy as subgraph isomorphism.

[96]This was due in part to the project leads, Chris and Matt Antoun, departing UC Santa Cruz after graduating with their masters degrees.

what has happened so far. For example, when two characters flirt, they are added to a register that enables a future tryst between them. Likewise, when a series of emergent actions results in a character adopting a motive to kill someone else, a register tracks this, which unlocks the possibility of that character murdering the prospective victim later on.

While these registers are used in a *forward* direction to structure the emergent possibility space, they could also be used in a *backward* direction to enable story sifting. To do this, the system would essentially chain backward across the registers, starting from the murder: retrieve from the motive register the entry that groups the murderer, victim, and motive, then proceed to the tryst register and do a similar lookup before likewise chaining backward to the flirtation register, and so forth. In some cases, it might not be possible to chain from one register to the causally preceding one, but this can easily be rectified by including the linking actions in the entries that are added to the registers—that is, as the simulation is unfolding, add into these registers the information that will be needed to chain backward during story sifting after the simulation concludes. I call this task *causal bookkeeping*. In Klein's system, this would mean adding into each entry in the murder registry a pointer to the flirtation-register entry that enabled the former, and so forth for all such contingency links.

Even if Klein's system could utilize causal bookkeeping to do story sifting, simply recounting an excavated kernel sequences in chronological order may not produce a good telling. For one, presentation is incredibly important, as I discuss in the next section, which means that bad prose will ruin otherwise good stories.[97]

[97] Will good prose save otherwise bad stories? I want to explore this notion in a study. My idea: have great writers write short stories that recount the most boring *Tale-Spin* outputs, and see how readers rate them relative to the originals. It would also be interesting to explore the inversion of this effect by rendering a beloved story in the style of a *Tale-Spin* output and seeing what people make of it. In a 1982 paper, Meehan himself acknowledged this dimension (following Natalie Dehn's emphasis on the author [252]): "In artificial intelligence terms, what we have

More immediately, though, the *arrangement* of the selected material in the actual telling is also critical. Here, we encounter the classic distinction between *fabula*, the true temporal ordering of selected events, and *syuzhet*, the ordering of those events in the telling.[98] As established above, the fatal flaw of Klein's generated murder mysteries is that they are not mysteries at all, because the identity of the murderer is revealed as the crime is transpiring. In a proper mystery story, there is divergence between the fabula and syuzhet such that the mystery is resolved later in the latter. Such divergence requires a decoupling of fabula and syuzhet, which in the case of emergent narrative requires a decoupling of simulation and telling. In Klein's system, however, the story is literally told *as it is happening*: the rules for natural language generation define which events should be recounted, and those events are recounted *as they occur*. Thus, the telling is actually a kind of simulation trace, but it does not recount every single event that occurs, so it is not monolithic like the extreme case discussed in the last section.

There are two solutions to the problem of introducing suspense into Klein's generated stories; one of these is specific to his project and the other is a general solution. The first solution, by which Klein's system could be "fixed" in less than an hour, is to simply change the rules for generating the natural language that expresses the actions that make up the recounted murder scenes. Recall that prior to a murder occurring, it is likely that several characters will have adopted motives for killing other characters. Thus, when a given character is murdered,

here is a domain problem. We are trying to reason about a form of writing, without having much expertise in the art of writing, and I expect no more success than I would from someone with no medical knowledge who tried to build a diagnosis system. The [story generation] literature reveals little or no contact with the practitioners, namely writers" [824, p. 459].

[98]In the case of historiography and curationist emergent narrative, we might also identify a third layer, which is the ordering of all events that occurred in the world, regardless of whether they were selected for telling. In the case of conventional fiction, this distinction is coextensive with the fabula, since the only events that can be known to have occurred in the fictional world are the ones that are reported in the work. This distinction is another argument for why emergent narrative works like nonfiction.

it is likely that multiple other characters will have adopted a motive for killing that person. As Marie-Laure Ryan notes, "a good mystery story should implant the false clues of possible, but unrealized explanations" [1065, p. 515], and this is exactly what the earlier scenes do.

Since there is no exclusive connection by which the murderer can be identified given a victim (which would be the case if only that character had a motive for killing the victim), suspense could be generated simply by rendering the murder scene without revealing the murderer's identity. This could be achieved by changing the rules for natural language generation (for the actions associated with the murder scene) such that identifying noun phrases are replaced with constructions like 'the killer' and pronominal expressions with non-gendered variants. Because the suspense depends on multiple characters having a motive to kill the victim, the simulation might also need to hold off on the murder scene until such circumstances hold for a potential victim. Similarly, Marie-Laure Ryan proposes a reformulation to Klein's system such that narration only begins after the murder scene, whereby the stories would be all about the detection phase [1065, p. 515].[99] This would take considerable work to implement well, however, due to the system's lack of rich detection modeling, as discussed in Section 4.1.5.

A more general solution to the introduction of suspense characterizes a broad approach to the task of curating simulated storyworlds: decouple the simulation and the curation procedure, such that curation occurs *after* the simulation has concluded. In the case of Klein's system, curation would begin once the simulation has concluded. This would entail the development of a second system that identifies the murder action as a most reportable event, and then chains backward and forward to excavate the larger kernel sequence containing the lead-up to the

[99]Similarly, Ed Kahn stated that a better telling could be achieved if a "global editing program [...] filtered out excessive randomness and gave more coherence to the story as a whole." [558, p. 5]. Below, I discuss how Klein himself actually intended for such curation.

crime and also its ramifications. This would result in a lot of extraneous material being excluded, which would generally help the prose, but it would also make sense to include satellites (to use Chatman's term) pertaining to red herrings such as non-murderers adopting motives to kill the victim. Such excavation of material beyond the kernel sequence indexed by the most reportable event is the real challenge of curation, since it requires an author to inscribe knowledge about (good) storytelling beyond the relatively simple matter of assembling a sequence of temporally and causally related events.

Klein Cared About Curation

Before moving on, I would like to clarify that Sheldon Klein was actually quite concerned with the idea of curation, which he referred to as "narrative style control" [601, p. 20]. In line with my proposal for automatic curation, Klein's *narrative style control monitor* was to reason not just about the style of prose, but also about the selection of material to recount: "The problem is what to describe", he aptly observed [601, p. 20]. In his first note on the project, Klein proposes that the narrative accounts could be limited to one character's experience:

> Design of the system includes the ability to follow just one character and what impinges upon him through the course of the plot simulation, or all characters simultaneously, or serially. [590, p. 418]

This selection technique, coupled with Klein's plans to generate first-person prose in targeted character voices [597, pp. 28–29], could have enable automatic *focalization*, which was not explored in story generation until several decades later [155, 700, 182, 847, 60].

Moreover, Klein writes lucidly about how it is necessary in any case to select a subset of simulated material for recounting:

> One might use the total semantic network to generate a total description of itself each and every time frame at a horrendous cost in time and redundancy. The first version of this program was set to issue descriptions of just the changes in the semantic network that occurred during the previous time frame. [...] To control the subject matter, it is necessary to have a powerful device for selecting particular subpaths through the semantic network to serve as inputs to the narrative [i.e., narration] generation component. Accordingly, a program is under construction for finding complex paths through the network that are a function of a variety of logical conditions. Typical requests to the program might be: "Find a path between node A and node B that passes through relation R_1, but not through R_2." [601, pp. 20–21][100]

Indeed, such requests could be used to isolate arbitrary subgraphs of the larger semantic network that Klein uses to represent the storyworld and its accumulated history. In this way, an author could inscribe patterns about what makes a good story. This expresses what I view as a strength of Klein's perspective, as opposed to the more or less contemporaneous *Tale-Spin*: Klein views story generation as a kind of latent authorship, while to Meehan it is an experimental method for doing cognitivist research that is essentially divorced from aesthetic concerns. Klein even mentions, humorously, that "the task of writing rules for generating a novel will probably involve as much work as the old-fashioned method" [601, p. 31].

Coining 'Curationism'

Thus, we find that the idea of curation is likely as old as the idea of emergent narrative: as I have shown, Klein described curation at some length, and his critics pinpointed poor curation as the fundamental limitation of his system. However, while these ideas have been in place for a long time, there has yet to be an extensive articulation of the act of curation as being *the central matter of importance in emergent narrative*, beyond even simulation. Such articulation is the fundamental

[100]A similar proposal appears in his other paper on the system [597, p. 28].

contribution of this dissertation, and it is why I use the neologism *curationism*: in my framework, curation is emphasized above all else.

4.2.4 Poor Presentation

As the previous sections show, practitioners of emergent narrative have often been more concerned with the generation of narrative raw material than with any subsequent curation of that material. One particularly salient form of curation error pertains to the quality of presentation of a narrative artifact; in the history of story generation, this has typically manifested as bad prose. Unfortunately, is not hard to find examples of this kind of deficiency.

***Tale-Spin*'s Stilted Prose**

Though it is not alone in this category, *Tale-Spin* is a prime example of poor presentation—in its case, bad prose—doing considerable harm to a project that is in many other ways quite impressive. This failing of the system is already well established (e.g., [124]), but let us briefly take a look at the opening excerpt of an example generated output:

> Once upon a time George Ant lived near a patch of ground. There was a nest in an ash tree. Wilma Bird lived in the nest. There was some water in a river. Wilma knew that the water was in the river. George knew that the water was in the river. One day Wilma was very thirsty. Wilma wanted to get near some water. Wilma flew from her nest across a meadow through a valley to the river. Wilma drank the water. Wilma was not thirsty. [822, p. 227–228]

This prose is clearly of poor quality, and for our purposes here it is not worth getting into the specifics, since this is not a thesis on natural language generation. Rather than discussing what makes this prose bad, I think it is more interesting to consider why this prose is bad. As Meehan makes clear in his descriptions of *Tale-Spin*, text generation was an ancillary concern in this project, which is primarily an exploration of character planning as a model for human cognition. Originally, Meehan had hoped to use a natural language generation system called *Babel* that was developed by another Roger Schank student, Neil Goldman, as part of his own dissertation work [404]. *Babel* unfortunately took up too much memory and was also missing a few features that Meehan required, so he decided to instead quickly hack together his own system, which he (aptly) called *Mumble*.[101] Here is how Meehan describes *Mumble* in his dissertation:

> It's a quick 'n' dirty program, written in a day, and it [sic] many of its parts do not correspond to the way humans speak, but it produces adequate, if somewhat verbose, sentences. Best of all, it's one tenth the size of BABEL. [822, p. 200]

Given the intellectual and technical contexts of its development, it is not surprising that *Tale-Spin* features bad prose—presentation was not a focus for Meehan.

Klein and the Cutting Edge

Another early project, however, features bad prose for a different reason. As noted above, Sheldon Klein was a pioneer of natural language generation who pushed its cutting edge throughout the 1960s and early 1970s [1158, 604, 587]. Thus, while the prose produced by his murder-mystery generator is of dubious aesthetic quality,[102] it actually represents the state of the art in natural language generation for its time. For instance, it was only in 1961 that Victor Yngve as-

[101]Meehan's *Mumble* is not to be confused with David D. McDonald's *Mumble*, which is one of the major early systems in natural language generation [815, 811, 812, 814], and one that is still in use today [813].

[102]The scholar Bernhard Lindemann characterized it as evoking "something like 'The New York Telephone Directory Retold and Edited by Gertrude Stein'" [689, p. 5]—that, however, sounds like an intriguing work to me!

tounded the nascent world of artificial intelligence by generating random sentences that were syntactically well formed, such as this one:

> When the water under trains has oiled wheels, four polished, shiny, little, oiled and little fire-boxs [sic] and its black, polished, proud, big, black and oiled fire-boxs, Engineer Small keeps the four wheels and fire-boxs black, big, little, oiled and heated. [1368, pp. 77–78]

By the early 1970s, when Klein published outputs from his system, the state of the art had not changed by much, and Klein was himself a leading figure in computational linguistics.

Thus, we might look a bit more deeply (and with more sympathy) at his generated prose, such as this excerpt here:

> James knew that Hume screwed Marion. James hated Dr. Bartholomew Hume. James wanted a revenge [sic]. James decided to kill Dr. Hume. [597, p. 108]

Note how the character Dr. Bartholomew Hume is referred to using three different noun phrases—this reflects effort on behalf of the system (and on behalf of Klein and his students as its authors) to feature lexical variation in the generated prose. In another excerpt from the same story, we find similar variation ('the policemen' vis-à-vis 'the cops') and also the impressive sentence "Florence also looked", which is a paraphrase of 'Florence looked for clues in the library' enabled by anaphora:

> The policeman arrive. The cops were idiotic. A detective examined the corpse. The cops looked for clues in the library. Florence also looked. [597, p. 108]

In fact, Klein designed his system to provide authorial control along several dimensions of curation (particularly with regard to presentation), which he identifies as including "lexical frequency, syntactic structure frequency, narrative subject matter, descriptive complexity, internal paragraph structure, paragraph grouping"

[601, p. 20].[103] Moreover, by this time Klein had already demonstrated the feasibility of exerting authorial control along these dimensions in what is perhaps the earliest work in natural language generation that emphasized authorial control of style [586, 587]. The problem, however, was that the specification of such control mechanisms would have been particularly burdensome in this era of computer programming. Thus, Klein's dubious prose does not reflect a lack of interest in natural language generation—he was a leader in the field and he situated the project as a contribution to that area—but rather they reflect the state of the art at that time with regard to the collected knowledge of its practitioners and moreover the hardware limitations of the era. As Klein explained in a later paper, the speed at which his system generated its prose was a technical achievement in that time period:

> Achievements with the generative portion of the system include a text grammar model that generates 2100 word murder mystery stories in less than 19 seconds each, complete with calculation of the plot and specification of the deep structure as well as the surface syntax. The speed of this generation is 100 to 1000 times faster than other existing programs using transformational grammars. (The algorithm for the semantics-to-surface structure generative component is such that processing time increases only linearly as a function of sentence length and syntactic complexity.) [592, p. 84]

Hardware Limitations and Emergent Narrative

While hardware limitations are no longer a major concern in natural language generation, it is important to consider them when discussing historical projects such as Klein's and Meehan's. More broadly, we still face limitations today in terms of working memory and computational efficiency, and these do constrain work in emergent narrative. A good example of this is the use of *level-of-detail*

[103] Section 5.1 of Klein's second paper on the system further emphasizes these architectural affordances [597, pp. 27–29].

modulation in underlying simulations [270, 741, 892].[104] In *Dwarf Fortress*, for instance, areas of gameworld are modeled more abstractly and simulated more coarsely depending on how far away they are from the player.[105] Another kind of approach in this vein pertains to *supporting character realism*: modeling supporting characters (typically in a real-time graphical world) in a computationally efficient way that does not break immersion [537, 1164, 332, 296, 946]. As Part II of this dissertation will show, I tend to eschew this technique in my own work, since I see it as a form of simulation tampering that inhibits the pleasures of emergent narrative outlined in Chapter 3, but this is at the cost of computational efficiency.

To return to Klein: while his system's generated prose is clearly of low aesthetic quality, this is not due to apathy or even to poor execution. Klein cared about presentation and he set out to produce prose of high quality, but it was simply not feasible to do so at this time. He could have decided to abandon the project early to work on something more near-term, but instead he forged onward to materialize the first widely reported effort in story generation. There will always be technical and epistemic limitations at play in the development of works of computational media. Practitioners simply have to work within these confines, but hopefully in doing so the confines will loosen. In considering historical works, especially computational ones, it is important to consider the particular limitations that would have constrained the development of the work in its time.

[104]Klein incidentally appears to have proposed level-of-detail modulation in a discussion of prospects for future work: "Other features include variable time scales for different rules and, accordingly, variable subdivision of the major time frame into subtime frames" [601, p. 32]. Curiously, Klein notes this about such modulation: "Interestingly, this feature creates problems that tempt one to assume Einstein's view of time and space; discussion of that topic is reserved for another paper" [601, p. 32]. Unfortunately, to my knowledge such a paper never appeared.

[105]Personal communication with Tarn Adams, June 15, 2018.

Emphasizing Emphasis

When it comes to presentation, I think it is probably most rewarding to consider how much a creator appears to have cared about it, and how much effort they exerted in attempting to produce artifacts that are presented well. Jim Meehan did not care about presentation, and the prose generated by *Mumble* reflects this. Sheldon Klein, on the other hand, did care about presentation, but unfortunately only a deep examination of the prose (informed by historical context) reveals this emphasis.[106] Some projects in story generation have been concerned entirely with presentation, or more specifically, with generated story prose. To my knowledge, the earliest example of this kind of effort is Paul Juell's forgotten 1981 dissertation, in which he seeks to improve the quality of prose for stories generated by Klein's system, *Tale-Spin*, and *Telltale* [554].[107] For instance, here is an example of *Tale-Spin*'s default *Mumble*-based output:

> Once upon a time George ant lived near a patch of ground. There was a nest in an ash tree. Wilma bird lived in the nest. There was some water in a river. Wilma knew that the water was in the river. George knew that the water was in the river. [822, p. 237]

For comparison, this is Juell's system's improved prose for the same passage:

> Once upon a time George Ant lived near a patch of ground. There was a nest in an ash tree and some water in a river. He knew that the

[106] Noah Wardrip-Fruin has called the practice of studying works of computational media with attention to their underlying processes and historical contexts *digital media archaeology* [1312]. Note: I prefer the term *computational media archaeology*, since computational processes are not inherently relevant to the understanding of all digital media (e.g., a digital image or a digital audio file).

[107] Like Ed Kahn's 1973 PhD thesis [559], Juell's Ohio State dissertation is also among the earliest in the field of story generation and it has likewise been forgotten to time. Interestingly, I recently encountered a 1993 post to a newsgroup in which Mark Kantrowitz, who worked on the Oz Project [563, 566], references Juell's dissertation as one of the few prior efforts in stylistic natural language generation (a tradition that he connects back to Sheldon Klein's 1965 paper "Control of Style with a Generative Grammar" [587]) [564]. Indeed, Kantrowitz's massive bibliography of natural language generation [565] is one of two documents to cite Juell's dissertation, the other being a more famous Ohio State thesis: Brenda Laurel's [654, p. 144].

> water was in the river. Wilma Bird knew that the water was in the river. She lived in the nest. [554, pp. 176]

Later efforts to improve the prose of story generators (or target particular pragmatic effects) include Ed Hovy's *Pauline* [513, 514], Mark Kantrowitz's *Glinda* [563, 566], Mark Lee's unnamed 1994 system [672], Clark Elliot's *story-morphs* project [308], Charles Callaway and James Lester's *StoryBook* [155], Nick Montfort's *Curveship* [847, 849] and *Slant* systems [851, 853], and Stephanie Lukin's *Fabula Tales* [712, 716, 713].

Presentation Modalities Beyond Prose

Finally, I would like to note that narrational prose is not the only presentation modality that is possible for emergent narrative. Later on, I will discuss a number of potential alternatives, ranging from physical installations to enacted performance to conversational speech to videogame quests to comics and more. Interestingly, some of these modalities appear to be easier to work in, in terms of the amount of limitations placed on the aesthetic quality of the presented artifacts. In any case, however, constraints are placed on each modality, and a lack of emphasis on presentation will doom works of emergent narrative regardless of the particular medium that is targeted for a curated narrative artifact.

Closing

To summarize the remarks of this section, poor presentation in earlier projects seems to have often resulted, naturally, from a lack of emphasis on the presentation aspects of a project. That being said, even when presentation is stressed, technical and epistemic deficiencies may limit the potential aesthetic quality of a presented artifact, and it is especially important to keep this in mind when considering historical works in the area of computational media.

4.2.5 Failure to Mount

One aspect of presentation is important enough to merit its own section, especially considering the degree to which it has contributed to the poor showing of emergent narrative to date. In this dissertation, I assume that projects in emergent narrative are more broadly concerned with media, which means that the practice is really about developing media works.[108] A major shortcoming of the work in this area, particularly projects that have come out of academia, is that its practitioners do not seem to emphasize the construction of actual media artifacts. That is, even projects in which curated narrative artifacts may be generated fail to then *mount* these artifacts as (or in) full-fledged media works.

Stop Mounting in Academic Papers

As a fundamental example, consider how generated artifacts in story generation, even ones that are the result of some degree of curation, have typically been presented: prose that is included in a larger academic paper. This is a not an authentic (or at least compelling) media genre.[109] While I concede that there is a certain aesthetic appeal to generated stories that are printed in all caps on fanfold printout—this is how *Saga II*'s raw outputs appear, as Figure 4.5 shows—the attraction is probably rooted in a kind of historical fetishism. This is not really a real media genre, is it?[110] I contend that projects whose curated narrative

[108]As noted in the previous section, practitioners like Jim Meehan have not always been concerned with media. In those cases, the works do not even make it far enough to fail in the particular way described in this section, and thus they are not worth considering here.

[109]And I say this as one of the world's major connoisseurs of generated stories printed in academic papers.

[110]A potential counterexample here is Alison Knowles and James Tenney's *A House of Dust*, which is an early project in computer poetry that resulted in the publication of a chapbook that was distributed as fanfold printouts contained in plastic bags [605]. This kind of presentation is evocative by virtue of its emphasis on the materiality of the computer poetry of its era. That is, fanfold printout is the final form of the curated product as the result of conscious deliberation—a kind of aesthetic doubling down—not simply because fanfold printout is how computer outputs

artifacts are presented in this way fail to complete the process of media creation.

Klein's Research as Media Practice

Sheldon Klein realized this. In his first technical report on the murder-mystery generator, he wrote the following about his aims for the project:

> It is a rather long distance from [the] promise of the title to the fragmentary output just presented. The first author of this paper is an outspoken logical positivist, and accordingly, foresees, as the inevitable test of the system and the semantic model it embodies, the publication of a novel produced by the program, with the publisher remaining ignorant of the non-human origin of the material. [601, p. 30]

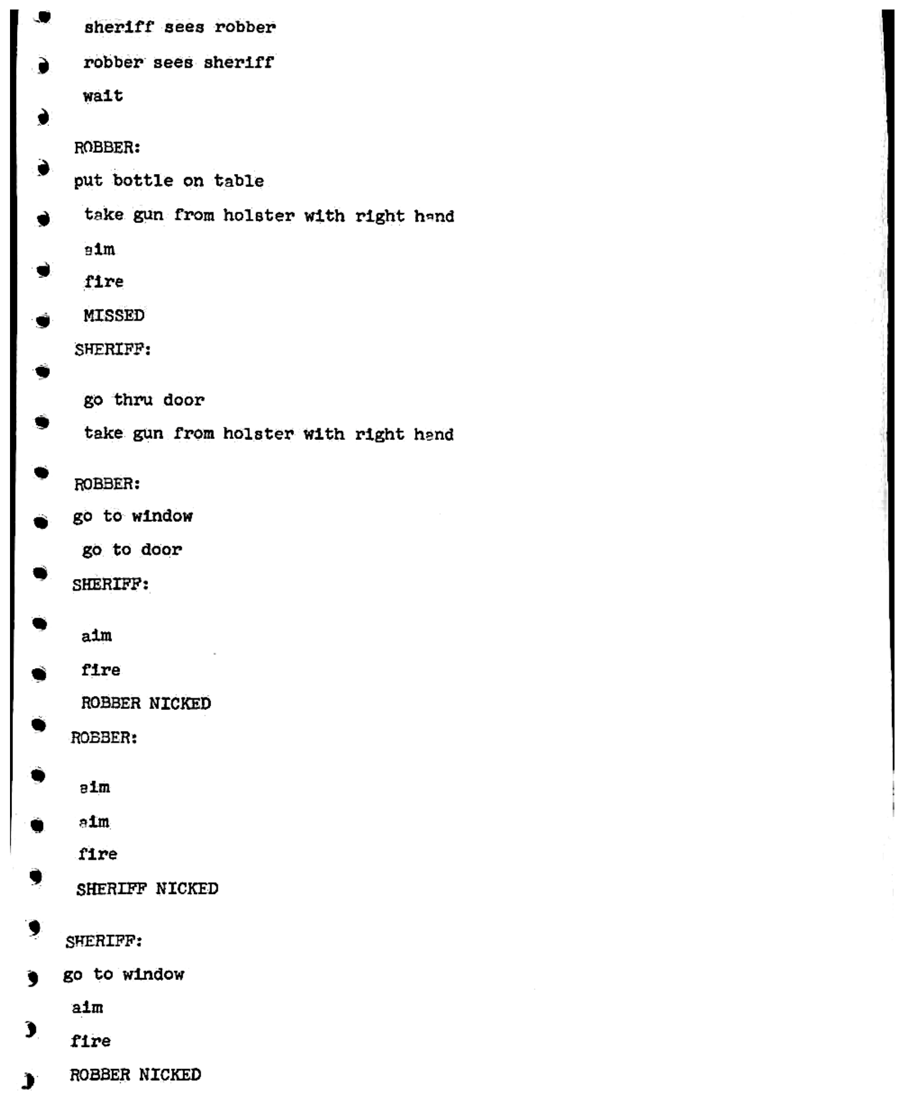
Klein articulates this goal using the language of scientific investigation, but I believe that this passage also suggests the importance of mounting generated outputs in full-fledged media works—in this case, a novel. After all, he calls his system an "automatic novel writer", not a 'story generator'.[111] To be fair, the latter phrase was not yet in use, but in any event Klein makes it clear that his project targets the specific media form of the novel.

Though his novel was never produced, it appears that Klein came close, as he wrote in the April 1974 SIGART Newsletter, "Results of our efforts should be available by the end of Summer 1974 in the form of a first draft of a book entitled, appropriately, The Machine that Ate the World" [558, p. 4].[112] While Klein's book never appeared, the system's outputs did appear in Jasia Reichardt's

were often experienced at that time. Moreover, the artifacts were distributed, which reflects an emphasis on culminating in a media artifact.

[111]Klein uses this phrase as early as 1969, in an interesting note on the project that reports a start date of September 1967 [590, p. 418]. In terms of the history of story generation, this is is considerably earlier than 1971–73, which is the period with which scholars who acknowledge the project have associated it. Still, it is not the earliest work in computer story generation, as I have already noted above: *Saga II* (1960) and a folktale generator by Joseph E. Grimes (circa 1963) preceded it [1039].

[112]Note that I call dibs on using this title at some point, potentially for my planned book on Sheldon Klein and his wild career.



```
sheriff sees robber
robber sees sheriff
wait
ROBBER:
put bottle on table
take gun from holster with right hand
aim
fire
MISSED
SHERIFF:
go thru door
take gun from holster with right hand
ROBBER:
go to window
go to door
SHERIFF:
aim
fire
ROBBER NICKED
ROBBER:
aim
aim
fire
SHERIFF NICKED
SHERIFF:
go to window
aim
fire
ROBBER NICKED
```

Figure 4.5: An excerpt from an original *Saga II* printout. Because it renders every change to the storyworld in detail, this printout pushes the limits of conventional definitions of narrative. Note, however, that this output is merely the raw material produced by the simulation, not the final product envisioned in the project. It was always intended that CBS-TV professionals would curate a set of printouts into production-ready shooting scripts, which would themselves be used to film live-action television segments. Thus, while the narrative quality of this output is questionable, it is through an act of curation that a recognizable narrative artifact (shooting script) is produced and, finally, mounted in a fully realized media work (television production). (Courtesy Computer History Museum)

Robots: Fact, Fiction, and Prediction [990, pp. 70–72] and in the French literary magazine *Action poétique* [70, p. 33].[113] In that same SIGART Newsletter in which his forthcoming book is announced, Klein also alludes to another evocative project: "There is even work on a musical composition generator based on a narrative discourse generation model" [558, p. 4]. This thread, it appears, would evolve into an effort to generate the multimedia video opera *Revolt in Flatland*. Here is a description of the project that appeared in *Byte Magazine*:

> More recently, Klein says, he and his students have invented a program to write operas. Like the earlier two programs, the opera-writer is divided into two parts. A simulator generates the plot in a special symbolic language devised by the group, then translators turn the symbolic plot into words, music, and moving pictures of the action. [...] To facilitate display on a video terminal, the opera is based on Edwin Abbott's nineteenth-century fantasy *Flatland* about a two-dimensional world [817, p. 51]

In a brief 1982 note, Klein notes this about the project:

> We use the simulation model as a generator of operas, complete with textual, pictorial and musical output, all derived from a common se-

[113] A decade earlier, the linguist Joe Grimes had planned to publish a book of computer-generated stories, which he would have called *Grimes' Fairy Tales*, but it too never appeared [1039]. I have encountered evidence that *Saga II* cocreator Douglas T. Ross had several of the system's outputs compiled into a one-off bound volume, but I do not know if this artifact survives. Over the remainder of the century, at least three published novels would be marketed as having been written by computer programs: the obscure *Bagabone, Hem'I Die Now* (1980) by *Melpomene* [828], the famous *The Policeman's Beard is Half Constructed* (1984) by *Racter* [964], and the remarkable *Just This Once* (1993) by *Hal* ("as told to Scott French") [359]. While the former two works have been exposed as hoaxes [1117, 75], French's admitted collaboration might have authentically integrated significant computer contributions. In the book's introduction, French claims to have spent $50,000 on professional-grade *expert systems* technology [359, pp. vii–viii], which he used to author thousands of production rules that operationalized the style of writer Jacqueline Susann [134]. The book has been the subject of academic legal discussion [1296, 191, 1360, 73], and even a lawsuit: Susann's estate contended that the system plagiarized her and sued French for copyright infringement, but the parties settled out of court [699]. Just last year, an indubitable volume of computer-generated stories was published by Counterpath Press: Rafael Pérez y Pérez's *Mexica: 20 Years—20 Stories [20 años—20 historias]* (2017) [934], which compiles twenty tales produced by his *Mexica* system [931]. Finally, many volumes of computer poetry have been published over the decades.

> mantic source. The output is in the medium of a videotape recording, and is generally entitled, *Revolt in Flatland.* [602, p. 150][114]

I appreciate how this description is clear about the specific target medium, opera, as opposed to the nebulous 'story,' which enables the cop-out of merely printing generated prose in an academic paper. It would be ridiculous to claim that an opera has been produced by its reporting in a document, but doing this for a 'story' seems to command less scrutiny. Moreover, Klein does not just state a target medium, but also a target media format ('a videotape recording'). When it comes to this project, Klein appears to have achieved his goal: a *Revolt in Flatland* videotape was shown at an academic conference [817], and ultimately the work was included in the Central Opera Service directory of operas that were composed in the 1980s [171, p. 72].

Narrative Artifacts Are Not Enough

As others have articulated elsewhere, computational systems that are intended to support media experiences cannot be truly appraised except through actual implemented experiences of the kind that they are meant to support [762, 1200, 505, 610, 1316]. In the case of emergent narrative, it is possible for a project to feature a strong simulation and sound curation, resulting in compelling narrative artifacts that still disappoint, due to not being mounted in an actual media experience. This can be caused by its practitioners not having a clear sense of the medium that they are targeting (beyond 'story'), and of course it is the unfortunate case of all projects that do not quite reach the finish line of culminating in actual media.

[114] I do not know of any earlier examples of algorithmic scoring in a multimedia work. Later, the Oz Project spinoff company Zoesis produced *The Penguin Who Wouldn't Swim* (1999) [1380], an interactive drama (in the style of a Pixar short) featuring an adaptive score implemented by composer Michael Gordon Shapiro [1141]. The project was informally presented at the 1999 AAAI Fall Symposium on Narrative Intelligence, as Noah Wardrip-Fruin notes [1311, p. 329], and it was also briefly mentioned in a 2004 note by Bryan Loyall [709]. In a recent conversation, Michael Mateas called it the most impressive piece to come out of the Oz/Zoesis effort.

Before concluding this section, I would like to briefly note that the narrative artifacts created in projects in emergent narrative do not have to themselves be mounted as standalone projects. Indeed, there is the intriguing prospect of mounting the artifact as a component in a larger media experience. In *Bad News*, discussed below in Chapter 10, curated narrative artifacts are mounted in a larger installation experience as stories that are told conversationally by a live actor. Another prospect, which I expound upon in Section 5.3, is to feed the curated artifact back into the simulation that it recounts. For example, characters in a simulated storyworld could tell stories about that world that are enabled by curation, and the very telling of those stories could alter the world, leading to new kinds of stories, and so forth. Indeed, a number of exciting possibilities emerge when the critical mounting step is reached.

4.2.6 Aesthetic Posturing

As I write this dissertation, Google artist in residence Mario Klingemann is working to build neural networks that can generate human portraits in the style of the old masters. Writing about the project for Fast Company's *Co.Design* blog, Katharine Schwab notes, "The results of his attempts are often humorous, showing just how bad algorithms are at conjuring something as complex as a human face" [1118]. In my opinion, however, the outputs that are dismissed as humorously bad are actually more compelling than their strictly representational counterparts. Figure 4.6 shows examples of outputs that cannot be easily distinguished from the works that the model was trained on, as well as ones that eschew representational convention in visceral expressions of quirks in the underlying model. Even if one enjoys representational visual art, which I admit I tend not to, I maintain that the outputs that are indiscernible from the genre works they mimic lack conceptual intrigue. While Klingemann's preferred outputs conceal their procedural origin, it surfaces in the other examples through curious artifacts that reveal something about the underlying mechanisms.

Toward a New Aesthetics of the Computational

What is the point of generating artifacts that perfectly resemble ones created by humans? A generator that does that has the conceptual intrigue of a photocopier or a printing press—it is a machine for generic mass production. Computer art resembling the old masters is a magic trick whose allure will quickly wear off. Moreover, there is already a massive accumulation of human works, and appreciation of these works is often rooted in their human origins: humans enjoy the expression in a human-authored work of the particular will and worldview of the human author. If there is a choice between human-authored works and perfect facsimiles generated by computers, I think humans will almost exclusively prefer the former. Computer-generated artifacts that embrace a new aesthetics of the computational, on the other hand, are distinct from human works not just conceptually but also in terms of form. By leaning into their quirky aesthetics, these artifacts offer features that human works cannot.[115]

[115]I should mention that one reason for building such a machine is to deeply study a creative process. In this pattern of inquiry, a computer program that generates artifacts is viewed as computationalizing the human creative process for producing such artifacts. By inspecting the operation of the program, it may be possible to learn about the human process that it models. Additionally, the program may be developed to instantiate a particular existing theory of that creative process (from the arts, social sciences, or humanities, for instance)—this is called *operationalization* [63]—in which case the resulting computer program may also be seen as rigorously testing the given theory. While I think it can be valuable to operationalize outside theories as part of the process of media creation, I am personally skeptical that doing so can reveal the creative process (or even tell us much about the theory). This is primarily for the reason that it is easy to create generators that produce the same artifacts but that are not *isomorphic* with regard to their underlying processes. This means that a computer program that generates identically to a human may do so in a way that does not resemble the human's process. Moreover, it is possible (or inevitable) to operationalize theories such that the computer program's instantiation of that theory is not isomorphic with the actual theory, which means the program in fact tests the theory constituted by itself, as opposed to the theory that it was

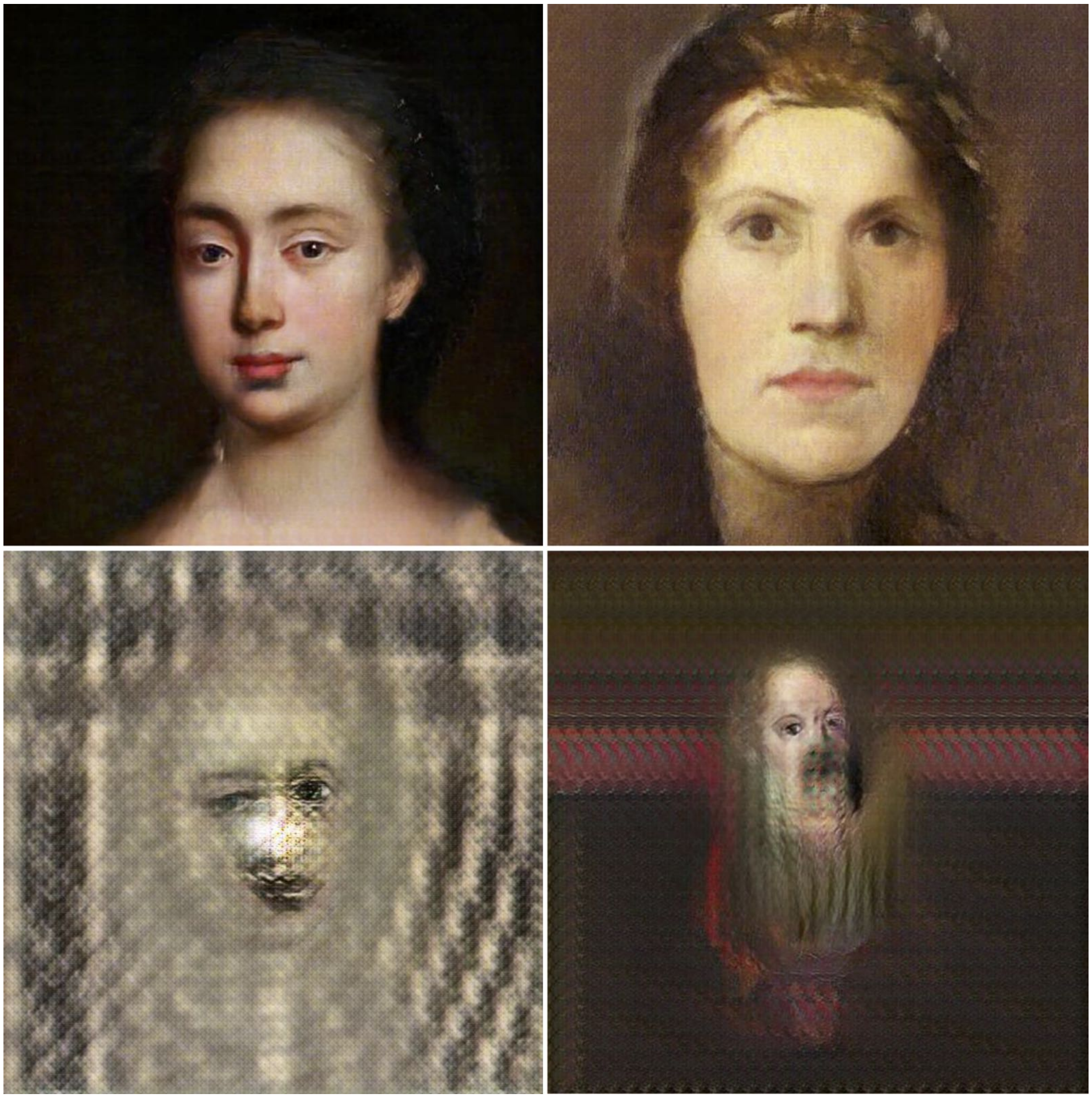


Figure 4.6: Images both "good" and "bad", generated by Mario Klingemann's system that is meant to match the style of the old masters. While images such as the bottom two have been dismissed as humorously bad, to me they are far more compelling than their strictly representational counterparts. The bottom images showcase an evocative computer-generated aesthetics, while the above two merely counterfeit a human sensibility. What is the point of generating artifacts that perfectly resemble ones created by humans? (Courtesy Mario Klingemann)

developed to test. Even if isomorphism obtains, it may due to a hyperassociative mapping scheme between source and target structure [675]. However, contemporary philosophers might argue that isomorphism is actually not important here, since the theory could be *multiply*

Eschewing Conventional Narrative Aesthetics

Analogously to the case I have just outlined, there is a tendency among practitioners of procedural narrative to target the aesthetic qualities of conventional narrative forms, but this aesthetic posturing can work to suppress some of the intriguing qualities that are uniquely germane to computational works. As Espen Aarseth writes in his seminal *Cybertext*:

> To achieve interesting and worthwhile computer-generated literature, it is necessary to dispose of the poetics of narrative literature and to use the computer's potential for combination and world simulation in order to develop new genres that can be valued and used in their own terms. [6, p. 141]

In emergent narrative, combination and world simulation are on full display, and this yields aesthetic qualities that diverge from conventional narrative works. In particular, the simulated storyworlds found in these systems work differently than the fictional worlds that are implied by conventional narrative works. While conventional works that are enabled by extensive worldbuilding efforts may feature a high level of detail in the terms of storyworlds that contextualize the stories recounted in the works, projects in emergent narrative go much farther by explicitly modeling the state of a storyworld and the mechanisms by which it may evolve. In making such commitments, quirks rooted in particular design emphases or

---

realized (realized in different ways) in its original context (scholarly writing, likely) and in the computational one (both being viewed as inscription technologies) [578]. Alternatively, others argue that even perfect isomorphism is not sufficient for meaningful equivalence [117]. These are murky issues in the philosophy of science [1221, 1286] and computational creativity [553], but thankfully the idea of operationalization as a mode of empirical investigation is not relevant in this dissertation: herein, I care about media creation and media artifacts, not the cognitive processes that drive creation or interpretation. That being said, I am still interested in two kinds of operationalization: the intuitive modeling of ideas from the arts and sciences and of tropes found in media, and the somewhat informal coevolution of a theoretical framework and a computational architecture, as articulated by Peter Mawhorter in his dissertation [788, pp. 16–18] (through the influence of Phil Agre [19]). As Part II of this dissertation will show, I make use of the former approach and the very subject of this paper, curationist emergent narrative, is a framework that has been derived by the latter method.

simulation abstractions may manifest as bizarre quirks in the generated surface material. When this happens, the system creator can work to rectify the underlying cause so that the quirk will no longer manifest in the surface material, but this becomes a Sisyphean nightmare—it is (probably) not possible to simulate a world that works just like our world, or even like the implied storyworld of any work of fiction. Moreover, in ironing out these wrinkles, the generated material loses the distinctive qualities that are uniquely present in works of emergent narrative. When this is done, to call back to Aarseth's message quoted above, the practitioner fails to value this new genre on its own terms.

***Tale-Spin*'s Secret Quantum Aesthetics**

Rather than viewing the idiosyncrasies of emergent narrative as wrinkles that should be ironed out, a practitioner might consider them instead as aesthetic qualities that should be amplified. In other words: why not lean into it? Let us consider *Tale-Spin* as a cautionary tale. As Noah Wardrip-Fruin argues in his chapter on the system in *Expressive Processing* [1311], everything in its simulation hinges on character plans:

> But *Tale-Spin* does not begin with a complete virtual world that characters can move across. Instead, movement happens only when required by plans, and the world is fleshed out only to the degree required by plans. Similarly, *Tale-Spin* does not generate a complete set of interpersonal relationships when characters are created but rather only fleshes out the connections between characters that are required by plans. [1311, p. 143]

The inclusion of character planning is certainly not unusual among narrative works, but in *Tale-Spin* its presence is amplified and warped through the mechanisms of its detailed simulation. In a typical narrative work, a character plans by considering at most a few candidate routes of action before selecting one course to take.[116] As such, stories that work in this way are still about the actions characters do take, which is the stuff of conventional narrative. In *Tale-Spin*, however, characters may consider every possible action, all of which may be unavailable to them given the state of the storyworld, and in this way the simulation becomes as much about character *inaction*. Characters spin threads of *possible worlds* [619, 921, 1066, 1022] in their minds, imagining what they *could* do, or what they would do, if the world were different. As Wardrip-Fruin puts it, by making it its operating principal, character planning in the system "creates the profusion of imagined worlds that define *Tale-Spin*'s fictions" [1311, p. 143].

Beyond characterizing its simulation, *Tale-Spin*'s strange enumeration of plausible alternate realities yields the core aesthetic intrigue of its narrative material. Stories are typically about paths that characters take, but the stories that emerge from *Tale-Spin*'s simulation are about paths that characters do not take, and in fact *cannot* take. Both aesthetically and conceptually, this evokes Jorge Luis Borges's exhaustive unraveling of the possible that characterizes works such as "The Library of Babel" and "The Garden of Forking Paths" [128]. It is a quantum aesthetics, or more precisely, an aesthetics of the quantum.

Frequently, *Tale-Spin* stories are tragic, but they are structured like strange *proofs of tragedy*: each is an exhaustive, well-structured articulation of all the preferable realities that are cordoned off from a character due to the cruel upshot of initial simulation conditions.[117] Wardrip-Fruin, for example, recounts the tale

[116] To be clear, here I am talking about narrative in a general sense, rather than computational narrative in particular, which means I likewise use 'character planning' in a general sense, as opposed to automated planning.

[117] Of course, *Tale-Spin* is capable of producing stories that are not tragic, but these seem to be less interesting and less frequently discussed. In one of his papers on the system, Meehan notes, "I have observed that people who run the program usually make the problem *very* hard; they find the resulting 'Trials and Tribulations' story more 'interesting' (their word) than 'Sweetness and Light' stories where all the characters like each other and do favors at the least suggestion" [823, p. 96].

of a hungry bear, Arthur, whose quest for honey was doomed from the start:

> And as we remember from the setup for the story, no one knows about any bear food in this world. Between Arthur and George they only know about one worm. So the story could not have turned out any other way. As soon as the audience decided that hunger was Arthur's problem, he was doomed. He made many plans, yet none of them had a chance of working. The end. [1311, pp. 136–37]

Given the initial state of the storyworld, there was no reachable later state in which Arthur Bear could acquire his sustenance, and the unfolding of the simulation simply proves this. The bear takes the only course of action available to him, and considers every unavailable course of action, and the resulting trace is an exhaustive and explicit proof of tragedy.

This is something strangely attractive about the alien narratology of a proof of tragedy. To be sure, it answers Aarseth's call, quoted above, "to dispose of the poetics of narrative literature and to use the computer's potential for combination and world simulation in order to develop new genres."[118] Unfortunately, however, the detailed operations of *Tale-Spin*'s mechanisms for character planning are elided in the surface text of its generated stories. Instead, its text generator *Mumble* recounts only those actions that characters actually take, rather than the rich mental simulation of actions that they would prefer to take. This is how storytelling works conventionally, but *Tale-Spin* narrative is not conventional: while character action is the stuff of conventional narrative, Meehan's worlds are structured around debilitating character inaction—the result of exhaustive mental

[118]Unfortunately, as Wardrip-Fruin explains in *Expressive Processing*, Aarseth did not recognize *Tale-Spin*'s peculiarity as exemplifying an evocative new poetics [1311, pp. 153–155]. This was likely due to a misapprehension of the system's underlying processes, which are heavily obscured in the surface text of its generated stories. This misapprehension is not unique to Aarseth's reading of *Tale-Spin*, but rather has been characteristic of much of the humanistic writing on the system. More broadly, the phenomenon of surface artifacts obscuring the richness of underlying systems is at play in many systems beyond Meehan's—fittingly, though, Wardrip-Fruin calls this the *Tale-Spin effect*.

consideration of all the promising actions that a character cannot take.[119] As a result of this unfortunate mismatch between the organizing principals at play in the underlying simulation and at the surface, the generated tales look nothing like proofs of tragedy, and *Tale-Spin* thereby renounces its own narratology.

The Wrong Narratology

Why did Meehan make this move? That is, why did he build a simulation about character planning—and as a byproduct, the consideration of possible worlds—and then curate this simulated material into accounts of character action? First, it is important to acknowledge that Meehan was far more concerned with the procedures of his underlying simulation than with the process of curating simulated material into a compelling narrative artifact. As I mentioned in the last section, *Tale-Spin*'s natural language generator, *Mumble*, was hastily developed in a single day. So one account of the mismatch between simulated material and surface narration is that *Mumble* is an incomplete system that was hacked together quickly, and thus does not represent an earnest attempt by Meehan to generate compelling narration. While this is almost certainly a contributing factor, I think it is still worth considering the particular way in which *Mumble* deviates from the narratological directives of *Tale-Spin*'s underlying simulation.

[119]It is worth noting that stories are also frequently about inaction, as Marie-Laure Ryan has compellingly argued: "narrative plots are layered entities, made up not only of a linear sequence of factual events, but also of the projections, wishes, plans, and interpretations produced by the characters as they reflect upon the world of which they are members. Insofar as they link events and states in a causal chain, these mental constructs present and structure a story, and may therefore be called 'embedded narratives'" [1064, p. 107]. What is wild about *Tale-Spin*, however, is the degree to which it models such embedded narratives: characters in fiction consider alternatives and project onto possible worlds, but they do not exhaustively consider *every possible world* with respect to a targeted goal. Thus, the unique aesthetic of *Tale-Spin* is its fiercely recursive *nesting model* of embedded narratives. It is all about what Gerald Prince has called the *disnarrated*: "events that do not happen but, nonetheless, are referred to" [959, p. 2]; unfortunately, while such events are referred to in the system's execution stack, they do not manifest in the generated prose.

The classic *Tale-Spin* storyworld is inspired by Aesop's fables, and I think that *Mumble* was designed to curate its simulated material into narrative artifacts that read like those stories. Thus, instead of a surreal account of Wilma unraveling a tangle of possible futures, we get sentences like "Wilma was inclined to lie to Tom" (see Wardrip-Fruin's illustrative account of this ellipsis [1311, pp. 144–145]). But *Tale-Spin*'s narratology does not work like Aesop's narratology—it is not even about character activity, let alone the moral ramifications or aphoristic suggestions of character activity—and so in targeting Aesop's style the system's generated tales fail to capture the interesting qualities of the underlying simulation. Moreover, because this underlying simulation does not work according to the principals that frame the narratology of Aesop, curation of *Tale-Spin*'s simulated material in the style of Aesop will tend to yield bad outputs. With a lot of work, *Tale-Spin*'s initial conditions can be rigged such that an actual Aesop tale is essentially reproduced in the *Mumble* generated telling—Meehan did this with "The Fox and the Crow" [823, p. 97]—but this does not mean that typical *Tale-Spin* runs will produce new fables that work like Aesop's.

More typically, the raw narrative material produced by the system, evoking a Borgesian hypersensitivity to the possible, loses much when it collapsed onto the rigid plane of actual executed actions. As such, the system's outputs have been widely ridiculed as boring stories, if not positively non-narrative. Indeed, Wardrip-Fruin has argued that the possible-worlds model implicit in its underlying processes (but missing in its surface) is perhaps the system's strongest connection to conventional modes of literature [1311, p. 142]. More charitably, we might interpret Meehan's project differently by assuming that his primary aim was to operationalize the narratology of Aesop's fable (in the sense outlined in footnote 115 of this chapter), but this does not appear to have been the case [1075]. If it was, moreover, Meehan's planning-based approach is like the *Rube Goldberg machine* [902] equivalent of a Proppian morphology [960]. Instead, we find the explicitly stated goal of modeling people [822, p. 4], though it is not clear whether this refers to human actors or human storytellers or both. In any event, as Wardrip-Fruin notes [1311, p. 151], its particular model of human cognition has since become dated. Wardrip-Fruin, however, argues that this makes the system even more interesting, since its narratology is a strange one rooted in the particular time and place of the Yale AI Project of the 1970s [1102, 1107, 250, 1165]. If only this aesthetic intrigue was reflected in the surface prose of its generated artifacts.

To One's Own Simulation Be True

The curation procedure of *Tale-Spin*, as instantiated in the operation of *Mumble*, serves as a cautionary tale about aesthetic posturing in emergent narrative. At the level of curation, works of emergent narrative should be true to their underlying simulations, so that surface artifacts may actually reflect the interesting features of the raw material out of which they are composed. Furthermore, practitioners should avoid curation modes that reflect different narratologies than those that are operational in their simulation procedures. This may confuse the notion of artifact evaluation with regard to such systems, but I believe that is the cost of doing business in a new medium.

Lovable Quirks and Maddening Flaws

Lastly, I would like to acknowledge that this section may appear to contradict the ones that have preceded it in this chapter. That is, in earlier sections I identified a set of undesirable project characteristics that should be rectified, while in this section I argue for the celebration of quirkiness. As an example, one could view the tendency of Klein's murder-mystery generator to reveal the killer halfway through its generated stories as a quirk on the order of *Tale-Spin*'s deep dive into character planning. The difference, critically, is that one of these features supports the goals of a larger project, while the other does not. Klein is attempting to model the structure and tropes of a particular genre, the murder mystery, and in that genre the killer is not revealed at the time that the murder is first recounted. Thus, the revelation of such information at the wrong time may be viewed as an error with respect to the goals of the project. *Tale-Spin*'s obsession with character planning, on the other hand, is in line with Meehan's goals for the project—in fact, this obsession *is* the goal of the project. Moreover, the quirkiness of *Tale-Spin*'s rummaging in possible worlds is evocative, while Klein's system's premature revelation simply detracts from the intrigue of its generated artifacts.

Of course, this is just my opinion, and there is no objective distinction between quirks that are compelling and ones that are not. Judgments will vary according to audience tastes, and also according to that of creators. In the end, it is up to a practitioner to decide what is best for her project, but I hope my argumentation in this section has succeeded in repositioning certain characteristics of emergent narrative (and generative art, more broadly) that tend to be viewed as negative as being potentially, or even inevitably, positive. I would like to see more generative art, but like Aarseth I think the genre will only find its place when the works that exemplify it begin to embrace, and conventionalize, new aesthetic principals.

Chapter 5

Refining Emergent Narrative

I will now present the crux of this dissertation, my framework for *curationist emergent narrative*. My aim in developing this framework has been simple: maintain the pleasure of emergent narrative (see Chapter 3) while simultaneously alleviating the pain (see Chapter 4). In meeting this goal, the framework takes an intellectual position, advocates for a particular technical approach, su ggests a general architecture, and produces a series of additional concepts. In this chapter, I will discuss these concerns in turn before proceeding to situate the framework against the intellectual contexts in which it was developed. Because many of the pertinent concepts and arguments have been discussed at length in the preceding chapters, in this chapter I will be relatively concise.

5.1 Intellectual Position

If we were to bifurcate the design space of procedural narrative, for our purposes here a convenient (though admittedly reductionist) policy would be to distinguish works that explicitly model a storyworld from ones that do not.[1] In this way, one sector of the bifurcated space hosts early physical devices like *Movie-Writer* [118] and *The Uranium Shipment and the Space Pirates* [856], both which were discussed in Section 4.1.2, as well as story grammars that operate only at the level of a story's surface text. Additionally, we would find here more technically involved efforts in story generation that focus specifically on language, such as *Pauline* [514], *StoryBook* [155], *Say Anything* [1227], and *Fabula Tales* [713]. These projects live in the first sector because they do not explicitly model a storyworld, even though the surface material that each generates will suggest the presence of a storyworld in the same way that the words of print fiction suggest the backdrop of a larger world.

Residing on the other side of this divide, then, are the projects in procedural narrative that explicitly model storyworlds. Here, we find simulationist story generators like Sheldon Klein's murder-mystery generator [601, 597] and *Tale-Spin* [822], as well as nearly all videogames that produce narrative, including examples like *The Sims* [792] and *Dwarf Fortress* [17], but also *The Elder Scrolls V: Skyrim* [1208, 867], titles in the *Football Manager* series [984, pp. 218–219], and countless other examples. We also encounter in this sector a number of narrative games that have been produced in academic research contexts, including *Façade* [775], *Prom Week* [803], *Best Laid Plans* [1318], *Nothing For Dinner* [438, 1235], *Emma's Journey* [1216, 1217], and *Bad News*, the subject of Chapter 10. All of these projects explicitly model a storyworld, and as such, this portion of the bifurcated design space coincides with the purview of this dissertation.

[1] I use the term 'procedural narrative' as a catch-all for story generation, interactive storytelling, videogame narrative, and any other modality in which narrative obtains through the mechanism of automatic procedure (potentially, but not necessarily, incorporating interactivity). Note that this term is more general than 'computational narrative', since it admits mechanical devices for story generation that precede the electronic digital computer.

Emergentism versus Interventionism

Within the sector containing works of procedural narrative that explicitly model storyworlds, we might carry out another convenient bifurcation, this time corresponding to fundamentally distinct intellectual approaches to the relationship between storyworld and story: *emergentism* versus *interventionism*. More precisely, we should say that this delineation is not binary, but rather works according to a continuum whose dimension captures the degree of interventionist mechanism that is employed in a work. To explain, let us consider the distinction in terms of a classic design challenge in this genre of procedural narrative (again, projects in which a storyworld is explicitly modeled).

The emergentist approach works bottom-up by modeling storyworlds from which stories may *emerge*, through the happenstance of simulation. This method characterizes some of the earliest works in computational narrative—*Saga II* [859], Sheldon Klein's murder-mystery generator [597], *Tale-Spin* [822]—and it is of course the hallmark of emergent narrative. As I argued at length in Chapter 3, the *pleasure* of emergent narrative is rooted primarily in its correspondences to the genre of nonfiction: when events emerge out of simulations, they feel like they really happen, and this unlocks the set of aesthetics that I outlined in Section 3.2. In turn, however, we experience a tandem *pain*: the commodity of storyworld simulation (raw emergent material) tends to lack story structure.

Given such a predicament, a potential remedy leaps out: *intervene* in the simulated storyworld to modulate the ongoing action such that it fits a model of story structure. This more top-down strategy characterizes several technical methods—author-centric story generation, drama management, plot-level narrative planning, and more—but I call the general scheme *interventionism*. Since Natalie Dehn's 1981 critique of *Tale-Spin*'s character-centric approach to story generation [252], the interventionist scheme has predominated in the field. In the area of interactive storytelling, it has been in place at least since Brenda Laurel's 1986 dissertation proposed a "playwright" module that could drive experiences toward dramatically pleasing outcomes [654].[2] The idea, again, is to intervene in (or just rig) the storyworld simulation so that story structure reliably obtains.

Killing the Pleasure with the Pain

While this remedy alleviates the pain of emergent narrative, it in turn kills the pleasure: when a simulated storyworld is modulated through the intervention of an external system—a model of creative writing, a drama manager, a plot-level narrative planner—it no longer works like nonfiction. By the interventionist pattern, events *spawn* according to the policies of a modulating system, which means they do not *emerge* out of the happenstance of simulation. They do not actually happen, and they do not feel like they actually happen. Thus, while interventionist outputs and experiences may be dramatically well-formed, they lack the distinctive aesthetics of emergent narrative that I outlined in Section 3.2. This does not mean that the interventionist approach is flawed, but rather that it is something altogether different from emergent narrative—it is not a remedy, since it kills the pleasure with the pain.

[2]Even earlier, when she was working at the Atari Sunnyvale Research Lab in the early 1980s, Laurel wrote a series of company memos that represent early articulations of her ideas for "interactive fantasy" [650, 651, 652]. A few years later, she compiled some of these papers in a form that she has made publicly available [655]. In one memo, Laurel and Eric A. Hulteen describe a planned experiment centered on a system called *Play-Right*, which was to adapt "portions of the design (if not the actual code)" of *Tale-Spin* [657, p. 2]. Later, Laurel published an extensive report laying out details of the proposed system, which would apparently operate in a *CAVE*-like [229] media room at Atari [653].

Desiderata for an Actual Remedy

An actual remedy for emergent narrative should maintain the pleasure while simultaneously alleviating the pain. It would remain fiercely emergentist, eschewing intervention, since the pleasure of emergent narrative depends on such inclinations. At the same time, however, it would produce story structure, since this alleviates the pain. As such, a remedy must follow a specific pattern: story structure emerges bottom-up, without intervention, in each simulation instance. If this were achieved, we could have our cake and eat it too.

This is the intellectual position that anchors my framework for curationist emergent narrative. It is fiercely emergentist, and thus strictly non-interventionist, because the aim is to have one's cake and eat it too. It is about maintaining the pleasure of emergent narrative while simultaneously alleviating the pain. In the next section, I describe the technical approach that makes this position tenable.

To Be Clear

An earlier version of this thesis demarcated emergentism and interventionism in binary terms, to which my committee members rightly took umbrage. Clearly there is something more like a continuum at work, where the degree of intervention increases as emergentism gradually gives way to interventionism. Even this is misleading, however, or at least imprecise.

For one, it is probably not possible for a simulated storyworld to come to exist *except* through a kind of human artifice—such worlds are created and evolved by rules and other modeling details that result from human authoring. As such, it does not make sense to consider a complete lack of intervention—where a storyworld comes to exist and evolve on its own—and so a pure emergentism does not seem feasible. Furthermore, there are projects that are surprisingly difficult to place on this continuum. For example, when we discussed this issue, my coadvisor Michael Mateas raised the case of *Façade* [775], which he co-created with Andrew Stern. While this project utilizes a well-known example of *drama management* [1010], and while drama management is a clear example of storyworld intervention, Michael explained to me that the project's drama manager actually works in quite bottom-up fashion—so much so that 'drama manager' might not be the most intuitive descriptor.[3]

In his feedback on my initial thesis draft, my other coadvisor Noah Wardrip-Fruin wrote, "it's ridiculous to think of the procedural script of Facade as an intervention in a simulation that would otherwise go on without it, or to think of the script as some kind of emergent simulation".[4] To be clear, I am not suggesting that the simulated world would go on by itself, but rather I mean to argue that such reliance on external mechanism expresses an interventionist inclination: if a system's simulation breaks down without intervention, the project is probably not emergentist. Nonetheless, I understand and agree with the broader point that my reading committee expressed to me, which is that my emergentist–interventionist delineation suffers under consideration of technical specifics.

As such, some clarification is necessary. Like I noted at the beginning of this dissertation, what you are reading is at its heart an art manifesto. It promulgates a fiercely emergentist approach to procedural narrative as an expression of an aesthetic orientation, as opposed to a line of technical or even intellectual argumentation. As we move into Part II, I will begin reporting on a series of projects that were developed in the context of my technical and media practice. This practice and the curationist framework that underpins it make sense to me because of the way I look at the design space: for me, the degree of storyworld

[3] Personal communication, July 26, 2018.

[4] Personal communication, August 10, 2018.

intervention *is* the central consideration in procedural narrative. While it may not be possible to build a purely emergentist system (because intervention is inevitable), I am still interested in developing projects that express an emergentist orientation and that target the peculiar aesthetics of emergent narrative that I outlined in Section 3.2. So, to be clear, my argument in this section is driven by an artistic inclination, more so than a technical or even intellectual stance. Even if no one else shares this inclination, I think it is important for me to explain it here, because it frames the various projects that are the core contributions of the second half of this dissertation.

5.2 Technical Approach

Let us return to the design challenge that I raised in the previous section: how can we can ensure that story structure reliably emerges across all the generable storyworlds of a simulation engine?[5]

***Dwarf Fortress*'s Narrative Cornucopias**

For a clue, let us look to an example of successful emergent narrative: *Dwarf Fortress* [17]. Tarn Adams is an emergentist—he eschews interventionist measures—but nonetheless we know from examples like *Oilfurnace*, the case study of Section 3.1.1, that story structure can still obtain in his storyworlds. Moreover, the abundance of published accounts of emergent stories [1287] suggest that story structure *tends* to emerge in *Dwarf Fortress*. Why does story structure tend to emerge here, but not in, say, *Tale-Spin*?[6] The answer is simple: the *Dwarf Fortress* simulation

[5] Readers who have read the earlier chapters of thesis will already have a sense as to how I answer this question. Nonetheless, I will reiterate the core ideas again here, since this chapter is intended to serve as a kernel for this dissertation.

[6] Of course, story structure can emerge in *Tale-Spin*, but typically this is the result of a human author rigging the system in a particular way, as Jim Meehan explains in his dissertation:

is far more complex—several orders of magnitude more complex. As such, the narrative potential of a *Dwarf Fortress* storyworld is much higher than that of a typical *Tale-Spin* world. Moreover, its simulation is so complex, and its narrative potential so immense, that it would actually be surprising if story structure did *not* emerge in a given simulation instance.

Thus, we might characterize Tarn Adams's method for ensuring the emergence of story structure as following a broad technical strategy: generate such an abundance of fertile simulated material that story structure is essentially *guaranteed* to be supplied in the accumulation. This strategy entails the simulation of massive storyworlds that have many characters doing many things over extended periods of story time. As such, this *cornucopian* approach to simulation differs fundamentally from *Tale-Spin*'s: *Tale-Spin* simulates a story, while *Dwarf Fortress* simulates a world, in all its overabundance. While Meehan crosses his fingers in the hopes that a simulation trace will itself constitute a well-formed story, Adams rests assured that many good stories are lodged in his system's material.[7] Indeed, the myriad goings-on of a *Dwarf Fortress* storyworld are not intended as a story, but instead as an *accumulation containing many stories.* As such, stories are repositioned as the *byproduct* of simulation, rather than its sole manufacture.

"we may or may not get a story by simply watching the simulator run. How do you make it interesting? You fix it in advance. You rig the world so that if people behave rationally, they'll do some interesting things" [822, p. 108]. As such, Meehan's approach positions him toward the interventionist side of things, though his system is sometimes viewed as an example of the failure of emergent narrative. In any event, when I claim that story structure tends not to emerge in *Tale-Spin*, I mean so with regard to the space of possible storyworlds that it can generate: while authors may rig initial conditions to select a more narratively interesting world from this space, most instances lack such intrigue. This is not the case with *Dwarf Fortress*.

[7]This critique is unfair to Meehan, since the hardware limitations of his day would have precluded this massively simulationist approach, even had he been so inclined.

But Now Curation Is Necessary

While Tarn Adams's cornucopian approach to simulation may ensure (without resorting to interventionism) that interesting stories always emerge, it raises a major challenge concomitantly: how can such stories be discerned amid the backdrop of boring simulated material? Indeed, if the plenitude is not a story, but rather an accumulation that contains many stories, then an additional step is required for humans to encounter or experience the stories: *curation.* By 'curation', I mean a process by which some entity sifts through the accumulation to identify narratively interesting material, which that entity then assembles into the shape of a story. In the case of *Dwarf Fortress*, humans are clearly encountering emergent stories, which begs the question: who is doing the curation? The answer, in this case, is humans. Human players are curating *Dwarf Fortress*'s simulated storyworlds, either through a basic mode of mental narrativization, where the ongoing experience of gameplay is understood as a kind of story, or through a more elaborate construction of second-order media artifacts that are meant for consumption by others—this is what *Oilfurnace* is.

Automatic Curation

What if a computer program could curate? This would enable a simulation to sift through its own material to excavate stories for human encounter, or even to discern partial stories that are in the process of emerging. Such a capacity would constitute a kind of computationalization of the mental processes that enable human players to recognize stories that emerge during gameplay. In turn, we may consider a prospect that is potentially more powerful: systems that computationalize the kind of curation procedure that Tim Denee carried out to create *Oilfurnace*; Figure 5.1 suggests this prospect in relation to an existing project.

Figure 5.1: Story scenes generated by *Mexica*'s visual narrator. What if the curation procedure that Tim Denee employed to create *Oilfurnace* was carried out automatically by a computer program? A recent project by Rafael Pérez y Pérez and collaborators has explored the automatic production of sequential art that visually expresses generated narrative content. In the images shown here, which are taken from a paper on *Mexica*'s visual narrator [935], plot points are rendered in a compelling visual style. If the visual narrator's narrative source material was excavated from a simulated storyworld, the resulting artifact would be a work of curationist emergent narrative in the style of *Oilfurnace*—but one that would be created automatically.

This would enable the automatic construction of full-fledged media experiences that recount the emergent happenings of simulated storyworlds.[8]

Overgenerate and Curate

This is my vision for curationist emergent narrative, and in describing it I have suggested the essentials of the technical approach that underpins it, which I capture in a tagline: *overgenerate and curate*. In natural language processing, the phrase 'overgenerate and rank' is associated with a technique whereby many candidate outputs (or parses) are generated and ranked, before a top-scoring one is ultimately selected.[9] In the case of the kind of cornucopian emergent narrative

[8] It would be an interesting challenge to attempt to specifically operationalize Tim Denee's curationist practice. There is a growing body of work on generating comics and other *sequential art* [304], which could provide a head start in terms of technical approach [938, 33, 673, 935, 750, 749, 162, 902, 965].

[9] The technique seems to be attributed to three roughly contemporaneous papers: a 1998 paper on generation by Irene Langkilde and Kevin Knight, a 2000 paper on parsing by Michael Collins [199], and a 2001 paper on generation by Marilyn Walker and collaborators [1302]. In linguistics, the notion of overgeneration refers to cases where a grammar (such as the rules for word formation in a natural language [444]) is capable of producing unattested constructions—

that is employed in *Dwarf Fortress*, a simulation overgenerates causal sequences, nearly all of which will be non-stories or boring stories, but through a process of curation the interesting causal sequences may be identified. Critically, the curation process is not simply one of ranking, hence my modification to the 'overgenerate and rank' tagline. That is, by 'curation' I do not mean the mere *selection* of a subset of the material generated by a simulation, but instead I intend to imply a more constructive procedure of which selection is only one phase. To illustrate the distinction, I will rely on two analogous procedures: curation in cultural institutions and Hayden White's model of historiography.

The canonical sense of 'curation' is the work done by a curator at a cultural institution—say, an art gallery—in preparing a collection for exhibition. Certainly there is a critical step in this process that entails the selection of a subset of the works in the collection that will actually be exhibited, but this alone is not curation. The result of this process is a set of items, not an exhibit. To build an exhibit, the curator must also reason about *presentation*, with all its considerations of spacing, timing, framing, priming, aesthetic details, and so forth. Furthermore, because the art exhibition is itself a cultural form [21], the curator engages in the established practices of that form: she builds her exhibit so that it looks and works like an exhibit—her design considerations are informed by the history of that cultural form and perhaps current trends too. If she decides to innovate, that innovation is *relative* to the established practice and its history. It is this highly constructive sense of curation that I seek to harness for my purposes here. As such, this technique does not entail the mere selection of a subset of simulated material, but the curation of that material by a procedure that results in a considered

such a grammar *overgenerates*. As a basic concept in generative grammar, the concept dates at least to Pāṇini's ancient model [571]. To my knowledge, the notion in linguistics of ranking candidate constructions is most associated with *Optimality Theory* [958], which is a model of linguistic production as a procedure that optimizes over competing constraints.

narrative artifact, which it itself mounted into a full-fledged media experience.

By now I have alluded several times to Hayden White's model of historiography, which I first introduced in Section 3.1.1. In pursuit of a *strange loop* [496], I will quote my earlier explanation and make this dissertation cite itself:[10]

> First, the subject phenomena are captured, as they are transpiring, in a *chronicle* that records them by some method of inscription. While the chronicle itself is the result of a kind of curation, since it will never perfectly capture the subject phenomena (inscription is lossy), it is, for the historian's purposes, the raw historical record. That is, because it is the only documentation of the subject phenomena, it contains the only material that the historian may use to construct her account of that phenomena (though she of course may augment this, or adulterate it, with extraneous material). The chronicle is open-ended, with no narrative beginning (but rather the unceremonious onset of a process of recording that produced the record) and no narrative end (but instead an unceremonious termination of that recording process). From this chronicle, the historian crafts a *story*. This process entails the selection of a subset of the chronicled events, which may then be used to construct a discernible narrative structure in which some of selected events actuate *motifs*: *inaugural motifs* cue meaningful causal sequences that will culminate eventually with *terminating motifs*, and along the way *transitional motifs* signal abeyance in the causal sequence. Finally, the story is embedded in a particular *emplotment*—e.g., tragedy or comedy—which unlocks a set of rhetorical effects that support a targeted ideological stance. [1040, p. 71]

White's model is the blueprint for curationist emergent narrative—by analogy to the former, we may conceptualize the latter. First, what I call (in that quotation) the 'subject phenomena', White more specifically calls the *historical field*. In curationist emergent narrative, the external reality of White's model becomes the simulated storyworld, and his historical field becomes the narratable phenomena that transpire in the course of a simulation instance. Just as a chronicle serves as a

[10] I recently encountered this device in a 1995 paper: "[Elliott and Melchior, 1995] Clark Elliott and Ernst Melchior. Getting to the point: Emotion as a necessary and sufficient element of story construction. In AAAI Technical Report for the Spring Symposium on Interactive Story Systems, Stanford University, March 1995. AAAI, American Association for Artificial Intelligence. THIS PAPER." [309, p. 4].

record of the historical field, a system may record data about what has happened in the simulation so far. This inscription will likely be lossy in both cases, since memory constraints make it infeasible to maintain a perfect record of a simulation. Because anything that is not recorded disappears, as in the real world, this data store contains all the material that may be used in the construction of a story, which brings us to the next step.

As White explains, creating a story entails the selection of material recorded in the chronicle, particularly such that a set of narrational motifs may be actuated—this is a Whitean notion of story structure, but of course any narratological formalism could be operationalized in this step. The point is that the data itself is a not story, but rather a structured accumulation of material that may be used to construct an actual story.[11] Finally, since the rhetorical considerations of emergent narrative are not as charged as those of historical writing, White's notion of emplotment does not connect so strongly. Nonetheless, it is important to acknowledge that there is an additional phase in story crafting beyond the assembly of selected content into a narrative structure—thus, we might view 'emplotment' as referring in a general way to this phase. Moreover, as I discussed at length in Section 4.2.5, in curationist emergent narrative constructing a narrative artifact is not enough. Practitioners of procedural narrative do not aim merely to build artifacts, but moreover to produce experiences. I believe that curated stories should be *mounted* in full-fledged media experiences, and in the next section I discuss some possible mounting configurations.

And so we have found our remedy: by producing massive storyworlds that constitute narrative cornucopias, emergent story structure can be guaranteed, but it is only through a procedure of curation that such structure may actually

[11]Here, perhaps surprisingly or perhaps not, we find in White's ideas an essential critique of *Tale-Spin*'s approach: it calls a raw simulation trace, with no actuated motifs, a story. Just as a chronicle is not an historical account, a simulation trace is not a story.

be identified—or more precisely, constituted. Thus, unsurprisingly, curationist emergent narrative does not circumvent the hard truth of interventionism: it takes work to maintain story structure. Critically, however, this framework provides a way to do this work without actually intervening in the storyworld, and as such it maintains the pleasure of emergent narrative while simultaneously alleviating the pain. This was my criterion for a remedy.

Appropriating Interventionist Methods for Curation

Intriguingly, all the knowledge engineering (about what makes a good story) that drives the interventionist scheme may be adapted to this emergentist approach. For example, a policy by which a drama manager fits ongoing action to a targeted dramatic arc may be converted into a *curation* policy that enables a system to assemble simulated material into the desired arc. In each case, the system reasons about how story units (e.g., quantized beats or simulated events) may be sequenced to compose dramatic arcs, but in curation the task of deciding what comes next is recast as search in a space of *observed* sequences. Likewise, the machinery of narrative planning could be utilized, but instead of planning operators that may be executed to *invent* actions, the planner reasons over operators that match against observed actions (ones that actually occurred in the storyworld). Even author-centric story generation could be employed: the task of story invention would be recast as one of story recounting, with the automatic storyteller utilizing a model of creative writing to tell stories about what has actually happened in a simulated storyworld.

Indeed, by recasting intervention policies as curation policies, *all* of the craft of interventionist procedural narrative may be appropriated for curationist emergent narrative. A corollary here is that automatic curation will require human creators to inscribe just as much story knowledge as before, but critically these authored story patterns do not modulate the storyworld itself, but rather the recounting of that world. This is the only difference between interventionist procedural narrative and curationist emergent narrative, and it is thus the hallmark of the latter. Thus, while foregoing intervention does not make the work any easier, it does afford an approach that stays true to the emergentist artistic orientation. As I have noted several times by now, developing such an approach is the core aim of this thesis.

How It Worked All Along

Lastly, I contend that this framework is actually an account of how successful examples of emergent narrative have worked all along: a simulation guarantees narrative intrigue in each of its generable storyworlds, and in turn humans willingly do the curation that is required to make emergent stories experienceable. In addition to explicitly identifying this pattern (about which more soon), a contribution of this dissertation is to postulate its computationalization, whereby curation happens automatically.

5.3 Curationist Architecture

The technical approach that I have outlined in the previous section suggests a general architecture for curationist emergent narrative, with two essential variants: in a *feedforward* curation architecture, constructed stories are mounted in second-order media experiences, while in a *feedback* architecture, they are mounted back into the simulated storyworld itself; Figure 5.2 illustrates both of these variants. I call this a 'general architecture' because individual implementations may vary considerably, and moreover because these modules do not necessarily have to be disentangled computer programs (or even computer programs at all)—instead, I mean to demarcate conceptually distinct bundles of functionality.

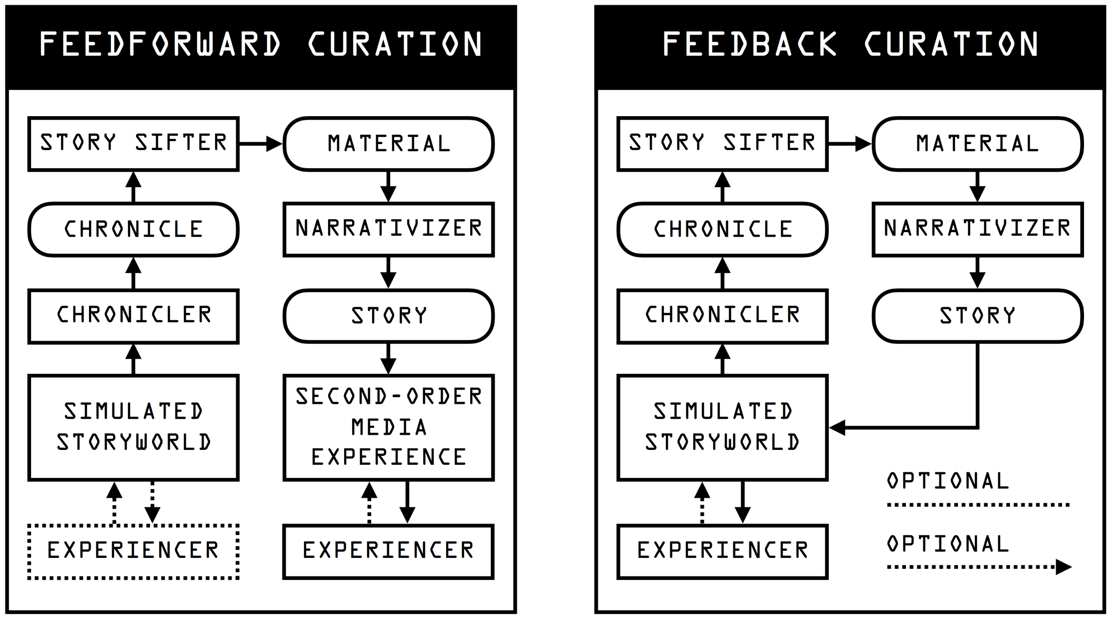


Figure 5.2: The two essential variants of a general architecture for curationist emergent narrative. In each case, the transpiring phenomena of a *simulated storyworld* are captured (by a *chronicler* module) into a *chronicle* that a *story sifter* module sifts through to extract *material* that a *narrativizer* module uses to construct a *story*, which is finally mounted in a full-fledged *media experience* that may be experienced by an *experiencer*. In *feedforward curation*, that media experience is decoupled from the simulated storyworld, making it a *second-order* experience. By *feedback curation*, the generated narrative artifact is mounted back into the simulated storyworld itself—this could occur, for example, in the case of a non-player character telling a generated story about events that transpired in the history of a gameworld. While the presence of an experiencer is optional with regard to the simulated storyworld—not all simulations are interactive or even observable—the media experience in which a generated narrative artifact will ultimately be mounted must be experienceable (though not necessarily interactive). Such optional patterns are indicated by dotted lines in the diagram.

Human Components

Before introducing the discrete components that work together to constitute this architecture, I would like to note that it does not necessarily have to take the form of a purely computational system. Indeed, as I have noted above, I maintain that this architecture actually explains how successful works of emergent narrative, such as *The Sims* and *Dwarf Fortress*, have worked all along: humans do the curation. As such, each module in this architecture may be human-powered, purely computational, or driven by a hybrid combination somewhere in the middle.

In a recent paper, Ben Samuel, Adam Summerville, and I explored this idea in the context of *computationally assisted performance* [1091]. Our process was to identify a series of roles underpinning both computational and noncomputational experiences (e.g., a *Hamlet* actor, a *Dungeons & Dragons* dungeon master, *Façade*'s drama manager) and then characterize each according to the degree to which it is: a performer of the experience, a consumer of the experience, computationally assisted, and endowed with agency. In positioning roles according to these four dimensions, we identified a series of tropes and some unexplored areas of design space, such as the development of games for AI consumption. For our purposes here, I would like to emphasize that the modules in this architecture may be human-powered, purely computational, or hybrid. Further, as I have stated above, in this dissertation I intend not only to characterize existing works of curationist emergent narrative according to this architecture, but to raise the challenge of building experiences in which conventionally human-powered modules are computationalized.[12]

5.3.1 Components

Without further ado, what follows are the individual components that make up the general architecture for curationist emergent narrative. Again, this archi-

[12]That being said, I am very curious about the prospect of doing the inverse: turning conventionally computational modules into human-powered (or hybrid) ones. I will discuss some ideas for this later on.

tecture is illustrated in Figure 5.2.

Experiencer

An *experiencer* is an entity who encounters, and optionally provides inputs to, a media experience. As I explain below, this experience may be a first-order one, meaning the simulated storyworld is itself encountered, or a second-order one in which simulated happenings are curated as narrative artifacts that are then mounted in an experience that is decoupled from the simulation. This component is the one most likely to be human-powered, though recent proposals have advanced the notion of developing media experiences for AI consumption—for instance, a role-playing game that a human develops for an AI to play [1091, 908]. In the case of curationist emergent narrative, I could imagine a human who is experiencing a simulation as a situated player character telling stories about simulated events to non-player characters who also live in the simulation.

Simulated Storyworld

A *simulated storyworld* contains the characters and events that an emergent narrative may recount. Conceptually, this module works like Hayden White's 'historical field': it is the world itself, not a record of that world. Architecturally, it is a simulation engine, though that may be entangled in the larger software ecosystem of an interactive experience. This module may seem the least amenable to human control, but there is a classical example: in tabletop roleplaying games, the storyworld is driven by a hybrid entity constituted in a human operator who administers the game's systems by which the world may be constituted.

Chronicler

The *chronicler* is a module that is tasked with recording the emergent phenomena that transpires over the course of a simulation instance. In Hayden White's model, it is implied to be a human who maintains a physical annal or chronicle, though in the real world there are also natural mechanisms that do this—for example, tree rings are a kind of chronicle. In a curationist architecture, this module may be human-powered, purely computational, or hybrid.

Chronicle

A *chronicle* is not a module, but rather a bundle of data. In a computational context, this may be constituted in data stored in memory or in a disk file, while in the case of human operation, the chronicle may obtain nebulously in a person's mind (about which more soon) or it may be recorded using analogue inscription technologies (e.g., written notes). As such, the chronicle is not necessarily contained in a single discrete record, but may be distributed across multiple records stored via multiple modalities.[13]

Story Sifter

The *story sifter* is a module that sifts through the accumulated simulated material captured in a chronicle to excavate raw narrative material out of which an actual story may be constructed. This component is the heart of the architecture, and the disregard for it in most works of emergent narrative to date is the impetus for this dissertation. In nearly all cases, this module has been human-powered—typically, its duties are carried out through a player's unconscious narrativization of an ongoing simulated experience—but I am calling for *automatic* story sifting.

[13] To be clear, by 'record' I do not mean 'database record' or anything else so specific, or so specifically digital, but rather an instance of any kind of storage medium, physical or otherwise.

To sift, this module must make use of policies that encode knowledge about what material makes for narratively potent material. Here, I differentiate between *sifting patterns* and *sifting heuristics*: the former pattern-match against the material recorded in a chronicle, while the latter encode abstract policies that may guide the sifting process. An example of a sifting pattern is a regular expression (encoding an abstraction of a type of storyline) that may be used to retrieve matching event sequences recorded in a chronicle. As for sifting heuristics, I could imagine a system that utilizes them working like Doug Lenat's system *AM* [674], which carries out heuristic exploration to automatically discover interesting or aesthetically pleasing mathematical concepts. In Chapter 11 I provide a detailed example of how these policies might be formulated.

Per my discussion in the last section, these methods may adapt all of the authored knowledge that is typically encoded in policies for story intervention or techniques for story invention. Indeed, story sifting is a full-fledged application area for which a multitude of methodologies could be developed. As I note below, in previous work I have referred to this module as the 'story recognizer' [44, 1058], but under the influence of Hayden White (who showed that stories are not already constituted in a chronicle) I now prefer the term 'story sifter'.

Material

The *material* excavated by the story sifter is another bundle of data, but critically it is a subset of the data included in the chronicle. It is the *stuff* of an emergent story, but it may also encompass material that will ultimately be extraneous (with regard to what is actually recounted in the constructed story). As I discussed in Section 4.2.3, this material may be highly structured—for example, a system's story sifter could build *story intention graphs* [312] that encode rich metadata about the relation between simulated events and other narratological concerns. The ultimate purpose of this material is to make it feasible for the narrativizer to build an interesting story.

Narrativizer

The *narrativizer* module constructs an actual narrative artifact out of the material produced by the story sifter. This task is as broad as it sounds, since it entails all of the work that is carried out in the application area of story generation. The difference between the narrativizer and a conventional story generator is that the latter *invents* stories, while a narrativizer *recounts* stories. Though it is currently obscure, the term 'narrativization' has a fairly long history in the humanities [1330, 920] and a technical sense of the term (building stories that recount data) is emerging in computational narrative [76, 1375, 1374].

Story

The narrativizer produces a *story*, by which I broadly mean a completed narrative artifact. Here, I use such a general term in an effort to be ecumenical—the story could take the form of prose, speech audio, a comic, or even something that is itself highly procedural, such as a videogame quest or a work of choice-based interactive fiction. Just as the narrativizer appropriates all of the methodology of story generation, and thereby all the craft of storytelling, the notion of 'story' in the curationist architecture encompasses all possible narrative forms.[14]

[14]To be clear, I do not mean to abscond from the duty of explicitly defining a specific notion of 'story'. As Hartmut Koenitz has argued, projects in computational narrative have tended to suffer from such vagueness [611]. Rather, I am being vague here (ecumenical, really), since I think it is critical to pin down such definitions at the project level, rather than at the level of a general scheme for procedural narrative, which is what I am articulating here. As such, I advocate setting such definitions as part of the act of grounding out this general architecture in a specific project-level implementation.

Media Experience

A *media experience* is a full-fledged media work that may be experienced by an experiencer. As I have now stated several times, and particularly in Section 4.2.5, it is not enough to generate a narrative artifact—for curation to culminate, that artifact must be mounted in a media experience.

Here, *Saga II* [859], which I discussed in Section 4.1.2, provides an illustrative example (see Figure 5.3). In this project, a raw script that is printed out by the TX-0 computer is what I am calling a chronicle, but critically the creators do not stop there and call that a final product. Instead, they use this raw printout to construct a shooting script—this is an act of curation that produces what I am calling a story. Still, however, the job is not done. As a final phase of curation, the creators produce a television program from the shooting script, thereby *mounting* the story into a full-fledged media experience. While this emphasis on curation is rooted in the peculiar impetus for the project—a collaboration between CBS-TV and MIT—it is remarkable nonetheless. I view *Saga II* as a shining example of curationist emergent narrative, though it would of course be even more impressive if the television program were itself automatically generated.[15]

As I will explain in more detail momentarily, the simulated storyworld may itself be a media experience—whether it is a full-fledged one depends simply on whether or not it may be experienced by an experiencer. In the context of curationist emergent narrative, the simulated storyworld (if it can be experienced) is a *first-order media experience*. If the generated narrative artifact is then mounted into an experience that is decoupled from the simulated storyworld, it has been mounted into a *second-order media experience*. As the project's sole technical

[15] It would be an interesting experiment to computationalize the entire *Saga II* curation pipeline—this would likely require *generative choreography* [113, 45, 225] and *generative machinima* [313, 864, 546]. Liza Daly's *Saga III* [236], a playful reimplementation of *Saga II*, could serve as a starting point.

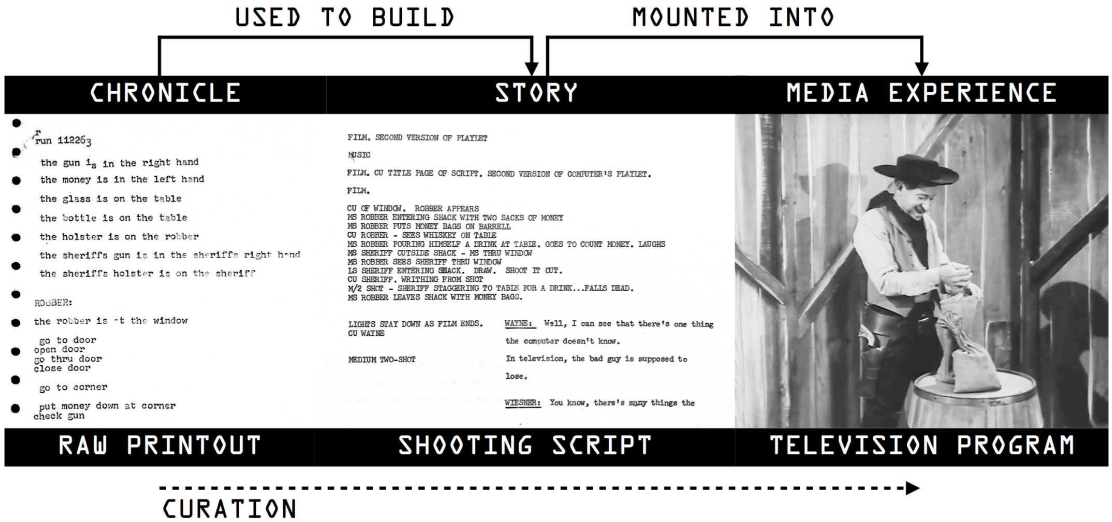


Figure 5.3: Curationist emergent narrative at work in *Saga II* (1960), a collaboration between CBS-TV television professionals and MIT computer scientists. The emergent phenomena of a simulated storyworld is captured in the system's raw printout (chronicle), which the television professionals curated into a shooting script (story) that was ultimately used to produce a television program (second-order media experience). Each step in this process represents a distinct phase of curation. In this case, the simulated storyworld and chronicler were computational modules, while the functions of the story sifter and narrativizer modules were carried out by humans. The major emphasis on curation was almost certainly due to the peculiar composition of the creative team behind the project, but nonetheless it is a shining example of curationist emergent narrative. An interesting technical challenge: fully computationalize the *Saga II* curation pipeline to generate 3D animations in the style of the original television scenes. (Courtesy Computer History Museum and Don Knuth)

memorandum [859] explains, the *Saga II* simulation did in fact have an interface that would allow an experiencer to modify the simulation rules. Given this, I would call the interactive simulation program a first-order media experience and the produced television scenes second-order ones. In that case, the experience in which curated narrative artifacts are mounted is not interactive, but of course it as also possible to mount into interactive experiences. This distinction is expressed by the optional arrows on the experiencer nodes shown in Figure 5.2.

5.3.2 Variants

Now that I have introduced these components, in this section I will explain how the two variants of this architecture are distinguished. Additionally, I will argue that an additional variant of the architecture accounts for how successful projects in emergent narrative have worked all along.

Feedforward Curation

By *feedforward curation*, a generated narrative artifact is mounted into a *second-order media experience*. In this way, the procedure feeds forward, curating material that is generated by a simulated storyworld for eventual presentation in a media experience that is decoupled from that simulation framework. An example of this is *Saga II*'s curation procedure, whereby events that emerged in a simulated storyworld (the modeled wild-west hideout) are recounted in an altogether separate media experience (a live-action television program).

Feedback Curation

By *feedback curation*, a generated narrative artifact is mounted back into the simulated storyworld itself. To my knowledge, there has been little exploration of this mode of curation, but one prospective example has been identified by scholar Marcus Carter in terms of what he calls 'emitext' (portmanteau of 'emergent' and 'paratext'): "a form of paratext that emerges from within the game as part of play, rather than a peripheral industry that surrounds it" [166, p. 311]. In curationist terms, this notion corresponds nicely to curated stories that are mounted in second-order experiences—for instance, the previously discussed *Oilfurnace*—but there is also another prospect that matches the feedback pattern. In his paper, Carter specifically identifies player-constructed propaganda in *EVE Online* [167], and such emitext can of course be presented to players during gameplay itself. When the propaganda is fed back into the storyworld in this way (to influence future emergence), feedback curation is at work.

Another example of this would be a videogame in which non-player characters automatically process the ongoing simulation to construct stories about what has happened so far. One interesting prospect for feedback curation is the emergence of stories about emergent stories—for example, a character's act of telling a story recounting emergent events could itself function as a critical event in a subsequent (higher-order) emergent story. This might termed *meta-emergent narrative*.[16]

I would like to clear up one potential concern. In Section 4.2.1, I identified a ubiquitous misconception by which the mere transpiring of simulation is believed to constitute emergent narrative, but now I am articulating a mode of curation whereby generated narrative artifacts are mounted back into a simulated storyworld. Is this the same phenomenon? No, these are not equivalent phenomena, for a simple reason: when the mere transpiring of simulation is treated as emergent narrative, the procedure of curation is circumvented, while in the case of feedback curation it is in fact the central concern.

Mental Curation

The curationist architecture can also account for how successful examples of emergent narrative have tended to work all along. By the pattern of *mental curation*, illustrated in Figure 5.4, the curated story obtains in the mind of a human experiencer.[17] As such, the various curation operations that are associated with

[16]Arthur Danto has evocatively suggested that historiography might be a necessarily cybernetic task: "the historian must make some more history before he can write some history, a distressingly sisyphian labor" [239, p. 153].

[17]In his comments on an earlier draft of this thesis, Noah Wardrip-Fruin asked, "In the mental curation diagram, shouldn't a bunch of those boxes move inside the experiencer box?" (personal communication, July 9, 2018). The answer is yes, that would be more precise, but the result

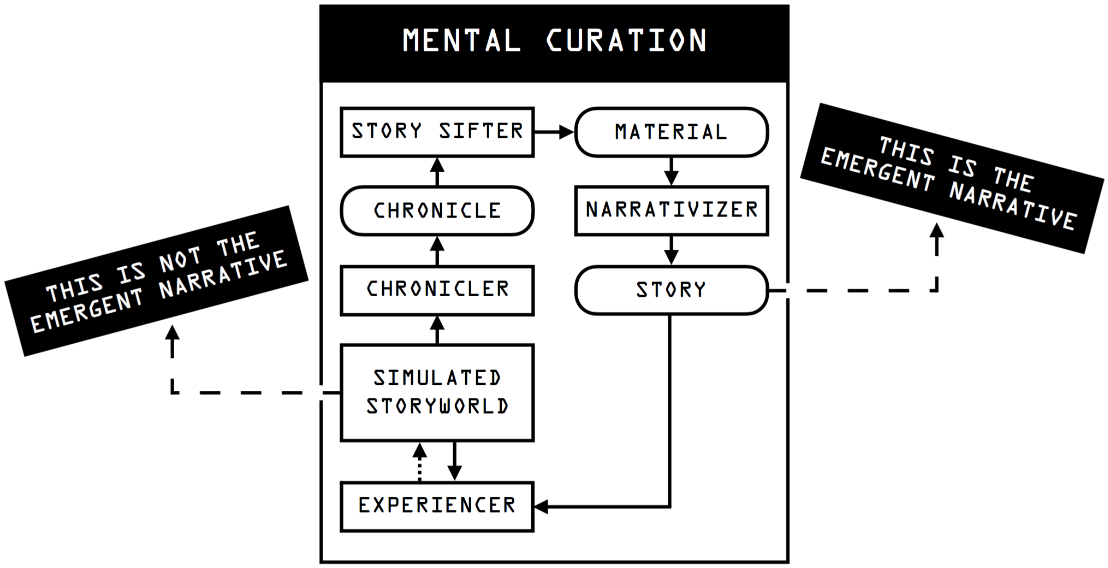


Figure 5.4: An illustration of mental curation in emergent narrative. The curationist architecture accounts for how successful examples of emergent narrative have actually worked all along: emergent stories obtain in the mind of an experiencer through an act of *mental curation*. Whenever an experiencer discerns story structure emerging out of a simulation, she has herself carried out the operations associated with the chronicler, story sifter, and narrativization modules. This configuration can be highly effective if humans are willing to take on these duties, but such disregard for curation has yielded a ubiquitous conflation of simulation and narration: it is often maintained that the transpiring of a simulation constitutes a kind of emergent narrative, but in actuality it can only manifest through a procedure of curation. This misperception relates to what Espen Aarseth has called 'narrativism' [4]. In terms of the curationist architecture, it represents a conflation of the simulated storyworld with an actual story. To be clear, not all human-powered curation is mental curation—the latter only occurs when emergent stories obtain in the *mind* of a human experiencer. It is possible for humans to curate external narrative artifacts that are mounted in actual media experiences—it is probably not worth calling mental life a media experience—but this configuration instantiates either feedforward or feedback curation (depending on whether the experience is the simulated storyworld itself).

would probably be hard to decipher and would not match the visual pattern of the earlier curationist architectural diagrams. In the later chapters, I will provide diagrams for each of my reported media experiences that show which architectural variant is instantiated in the project. In the case of *Diol/Diel/Dial*, mental curation is at work, and so I adapt this diagram (as Figure 8.5) to make it clear that, in that project, the experiencer is doing the work of the story sifter and narrativizer. Hopefully, if only in that case, the diagram is clear enough.

the chronicler, story sifter, and narrativizer modules are carried out by that human experiencer, likely as unconscious mental processes—that is, the experiencer cannot help but mentally narrate the events of a simulation as they are transpiring, or afterwards through reflection, and this narration procedure is an act of mental curation.[18] As such, mental curation may be viewed as a kind of degenerate variant of the curationist architecture.[19]

To be clear, I am not equating human curation with mental curation—the latter specifically occurs when a curated story obtains only in the mind of a human experiencer. It is also possible for a human to curate an external narrative artifact that may be mounted in an actual media experience, whereas it is probably not worth considering mental life to be a kind of media experience. When mounting is carried out, either feedforward or feedback curation is instantiated. For example, *Oilfurnace* was produced by feedforward curation, whereas

[18]In his recent dissertation on player-curated accounts of *The Elder Scrolls V: Skyrim* [1208] gameplay, Eric Murnane argues that emergent narrative can *only* obtain through a distanced postmortem reflection: "There is a narrative that occurs only when we are not playing the game. [...] There is the story that the game is telling and the blow by blow of what the player does. However, when the player has a bit of time to reflect, they start to make connections. The beginning of emergent narrative is the surprise. Something happens in the game that reminds them that it is a game. The player's suspension of disbelief is just shaken enough that the immersion is broken. Players can either reject the narrative outright when this happens, or (more interestingly), they can reflect on the event and contextualize it. If they take the latter route, suddenly all manner of things fall into place" [867, pp. 180–181]. I personally think that narration can occur in the moment, though certainly there is some latency, and in any event the most coherent stories will obtain through a more prolonged process of curation. Moreover, storytelling is typically predicated on its recounted events having already culminated (in an actual world or in an author's mind), and this requires a postmortem vantage; Arthur Danto's *narrative sentences* [239], discussed above, exemplify this idea. In finishing his thought, Murnane aptly identifies the cocreative nature of what I would call human-powered curationist emergent narrative: "I posit that the reason players do this is because they recognize that this surprise event is special. It is not what the developers intended; they are off the beaten path. When they make sense of the events in the game, they are becoming the coauthor. They are no longer receivers of media, they are cocreators" [867, p. 181]. Such cocreation is the crux of Ben Samuel's notion of *shared authorship*, which is the subject of his recent dissertation [1084].

[19]Here, I mean to use the technical sense of the word 'degenerate', which is to say that the curationist architecture is not fully instantiated in experiences that follow the pattern of mental curation. To be clear, I am not contending that such experiences are inferior in any objective sense, but rather that they do not demonstrate the particular architecture that I am promulgating with this dissertation.

human-powered feedback curation characterizes the phenomenon of *EVE Online* propaganda being mounted in the game's storyworld itself. Indeed, the prospects for human-powered curation are exciting, though so too are those of automatic curation. I advocate both, but in the context of a thesis in computational media, I will tend to emphasize the latter.

5.4 Additional Concepts

In this section, I will outline a few additional concepts that are pertinent to the curationist approach. Most of these have already been introduced and discussed in earlier parts of this dissertation, but I am including them here in an effort to develop a handy glossary of sorts. Readers may wish to consult this section if any of these concepts become hazy by the time they are discussed again later on.

Cornucopian Simulation

A *cornucopian simulation* is one that reliably produces storyworlds that teem with narrative potential. While the failure of emergent narrative to date has largely been rooted in a lack of story structure, this pain dissipates when every simulation instance generates narrative intrigue. In this regard, *Dwarf Fortress* is a paragon—each of its storyworlds is a narrative cornucopia, and this is because Tarn Adams is the consummate simulation author.

Bad Runs

In curationist emergent narrative, *bad runs*—simulation instances that do not produce narrative intrigue—are recast as authoring errors. This is a corollary of the emphasis on cornucopian simulation. While bad runs have conventionally been viewed as an essentially unavoidable pitfall of emergent narrative, in this framework it is the responsibility of an author to craft a simulation whose every instance bursts with narrative potential. If a simulation struggles to generate narrative potential, its author should consider adding more breadth (characters, entities, story time) and depth (complexity, richness, intrigue) to its modeling.

Contingent Unlocking

As I explained in Section 4.1.6, *contingent unlocking* is a technical approach to character simulation whereby emergent events may contingently unlock subsequent events. For example, in Sheldon Klein's murder-mystery generator [601, 597], two characters who emergently flirt unlock the affordance of a subsequent emergent tryst, which in turn unlocks pockets of the action space that pertain to the emergent murder scene. More typically, character simulations make events dependent on aspects of the world state, but this can lead to an overly realistic notion of causality (where an event's causality is distributed across essentially everything that has happened so far). Stories tend to cue clear causal relations between events, and when contingent unlocking is utilized, event structures may emerge that resemble the causal structures that are actually found in stories. I call the former *emergent contingency structures*, and contingent unlocking may be used to generate them and also to facilitate their identification later on, by enabling a task that I call *causal bookkeeping*. (These related concepts are reintroduced below.)

To be clear, contingent unlocking is not the only way to produce this kind of causal structure. I write about it in this thesis because it is a method that I have found to be useful in my own simulation practice (as I recount in Part II), and one that I find to be particularly compatible with the emergentist inclination, since it can guarantee that narratable event sequences will exist in a possibility space without enforcing their generation at any time. Moreover, while I will discuss a particular method of contingent unlocking that is employed in *Hennepin*, there are many possible implementations.

Causal Bookkeeping

Discussed at length in Section 4.1.6, *causal bookkeeping* is a technique whereby causal relations among emergent events are tracked by the system as they occur. This task is facilitated by contingent unlocking: whenever an event emerges by virtue of being contingently unlocked by an earlier event, the system records a causal relation between the two events. While a story is more than a sequence of temporally and causally related events, the tasks of story sifting and narrativization are greatly facilitated when this kind of information can be known upfront, rather than inferred later on. To be clear, here I mean to refer to a particular kind of causality (which features heavily in narrative): the *contingency* event relation, as identified by narratologist Mark Alan Powell [955, p. 40], whereby an event is caused by one or a few earlier events (and the connection is explicit and legible). As the following chapters will show, the evolution of my simulation practice over three engines has been marked by the development of increasingly better mechanisms for causal bookkeeping. Due to the nature of simulation, I view temporal bookkeeping as a trivial task, but it is of course just as important to keep track of when events occurred relative to one another.

Emergent Contingency Structure

An *emergent contingency structure* is an event sequence that emerges out of a simulation and exhibits the kind of explicit and narrow causal structures that one finds in narrative, as opposed to the massively branching structures that characterize how causality actually works in the real world and in simulations. This concept was discussed at length in Section 4.1.6.

Critically, an emergent contingency structure is built only according to *contingency* relations between events, which I have just mentioned. With causal bookkeeping in place, the identification of an emergent contingency structure is a simple matter of retrieval that works by operating over recorded causal relations between events. To be clear, an edge in such a sequence does not represent an event contingently unlocking another kind of event, but moreover a case of an instance of the latter event actually emerging by virtue of the unlocking. That is, contingent unlocking enables the emergence of a subsequent event, but does not guarantee that happening.

Rather than a linear sequence, an emergent contingency structure will likely be a directed acyclic graph with a small branching factor—this is because an event may contingently cause multiple later events or be contingently caused by multiple earlier events (working in tandem).[20] Briefly, I would like to emphasize that the bottom-up generation of emergent contingency structures does not work like the top-down authoring of branching plot graphs, even though both kinds of structures may take a similar shape. When an emergent contingency structure obtains, it is due to the component events actually emerging in a simulation. Antithetically to this, the branching structure of an authored plot graph obtains through human invention. As such, the former phenomenon works like nonfiction, while the latter works like fiction.

Finally, I would like to emphasize that the emergence of (semi)linear event structure does not preclude the designator 'emergent narrative'. Emergentism is not dependent on a lack of recognizable structure, but rather such lacking has

[20]These graphs are acyclic because an event cannot cause itself, but in certain cases a cycle might actually obtain. For example, in a plot featuring time travel, an event could paradoxically cause itself, and this would be represented by a cycle in the graph.

been a symptom of deficient simulation methods that have unfortunately come to characterize the approach of emergent narrative. All stories have discernible causal structure—that thing we call 'plot'—the production of which should be the aim for *all* approaches to procedural narrative, including emergentist ones.[21]

Sifting Pattern

A *sifting pattern* specifies an abstract protonarrative pattern that a story sifter may use to excavate material in a chronicle that matches the pattern. For instance, one could author regular expressions that do this. In a 2015 paper [906], Joe Osborn and collaborators applied his *Playspecs* tool [909, pp. 201–218]—which supports regular expressions for playtrace data—to *Prom Week* playtraces to excavate examples of observed event sequences matching a specific pattern (e.g., two characters date, break up, and then get back together). While Osborn and his collaborators were particularly interested in using such patterns for authoring support, *Playspecs* could also be utilized to specify curationist sifting patterns.[22]

A sifting pattern could even contain nested sifting patterns, thereby taking the form of a full-fledged *story grammar* [1034]. While story grammars have been devised to capture patterns attested in corpora or to generate stories, this would

[21]To be clear, not all *literary works* have discernible causal structure, but I think it is fair to say that all *stories* do. I love William S. Burroughs more than the next person, but this predilection does not lead me to view any old emergentist welter as compelling experimental literature. One *could* attempt to construct a Burroughsian simulation whose traces might be imbued with a Burroughsian quality, but this would be a monumental design challenge that would require deep consideration and conscious effort extending far beyond the mere crafting of an emergent possibility space. In fact, my friend Joe Krall and I planned to take on such a project—more specifically, a *cut-up* [153] massively multiplayer online game—but tragically he was killed in an accident.

[22]Joe Osborn is my labmate, and through our mutual coadvisor Michael Mateas, we descend academically from Stephen Cole Kleene, who actually introduced the concept of regular expressions in a 1951 technical report titled "Representation of Events in Nerve Nets and Finite Automata" [583]. In his original formulation, the concept was associated with a notation for defining what he called "regular events" [583, pp. 46–75]. Kleene advised a student named Robert Constable, who advised a student named Joe Bates, who coadvised a student named Michael Mateas, who coadvised students named Joe Osborn and James Ryan.

be a novel application of them to the task of *retrieving* potential story material. To me, a sifting pattern suggests the image of a *sieve* whose mesh pattern instantiates a particular protonarrative archetype—the story sifter pours the chronicle through the sieve and emergent examples that match the pattern are caught in the mesh.

Sifting Heuristic

A *sifting heuristic* specifies a policy for sifting, rather than an explicit pattern that may be matched against the chronicle. Here, I imagine a story sifter that works similarly to Doug Lenat's system *AM* [674], which attempts to discover mathematical concepts by utilizing heuristics about what makes such a concept interesting or aesthetically pleasing. This variant evokes the idea of an inspired story sifter that explores a chronicle improvisationally with ideas about what makes for interesting or aesthetically pleasing material. As such, to connect back to Hayden White's work, a story sifter that works heuristically more closely resembles an historian than a sifter that uses patterns. An example sifting heuristic might specify that statistically improbable event sequences are interesting. As I noted above, Morteza Behrooz and collaborators have explored this idea in the context of logs from a rummy videogame with chat interaction [96].

5.5 Intellectual Development

In the interest of providing intellectual context, I will conclude this chapter with a brief account of the development of this curationistist framework. As I explain more in the next few chapters, toward the end of my work on *World*, my first simulation engine, I became interested in the idea of automatically discerning stories that emerge from simulations. In February of 2014, I wrote about the prospect of doing this for *World* storyworlds in an email to Reid Swanson in advance of a scheduled meeting:

> Among other things, I'm interested in brainstorming about how to sift out the more interesting stories in the sim, and how to generate short biographies for people in the world without just using a template.[23]

My primary challenge at that time was rooted in my system's lack of causal bookkeeping (discussed in the previous section), which meant that causal relations among events needed to be inferred as a first step. I recall also having conversations about this technical challenge with Peter Mawhorter, a labmate in the Expressive Intelligence Studio, around this time.

Dwarf Grandpa

In the spring of 2014, Jacob Garbe, another labmate, was carrying out a likeminded project under the moniker *Dwarf Grandpa*. Jacob's aim was to produce a system, called *LegendsWriter*, that could sift through material produced over the course of a *Dwarf Fortress* simulation instance to generate stories recounting emergent events that had occurred in that storyworld.[24] As he explains in an unpublished class report [375] and a recent blog post [376], Jacob hoped to build a fully automatic system that would be driven by the data contained in a *Legends mode* [279] export file.[25] In curationist terms, the Legends data is a chronicle of a storyworld, and the *LegendsWriter* would do story sifting using sifting patterns that Jacob calls *story scaffolds*. As he explains in a recent blog post recounting the project, these are like regular expressions for event sequences (and with nesting, they become full-fledged story grammars):

[23]Personal communication, February 6, 2018.

[24]Just as grandfathers are known to tell stories about the real world, the system would be like a dwarf grandpa who tells stories about *Dwarf Fortress* storyworlds—hence the moniker.

[25]Legends mode provides an interface to the history of a *Dwarf Fortress* simulated storyworld. It allows a player to view maps and explore data representing emergent phenomena pertaining to characters, locales, events, and more, and much of this data may be exported as well.

> I made a small library of scaffolds. In this implementation, a scaffold was a grouping of sub-events that were ordered in time, but not in strict increments (i.e. I wouldn't care if "Eat" came 2 or 3 or 50 timesteps after "Hunt", I just cared if "Eat" came after "Hunt"). These scaffolds [...] could read simple data from state (like which characters were involved, etc). Once I had a library, I ran pattern matching over the events for each character in the exported Dwarf Fortress history, checking to see if I could apply a scaffold to it. [376, n.p.]

After sifting out raw story material through the employment of story scaffolds, *LegendsWriter* performs narrativization by using a grammar-based text generator that can produce story prose given the raw material associated with a given scaffold. According to Jacob's ultimate vision for the project, generated stories would be mounted in some kind of media experience (perhaps one in which an elderly dwarf character recounts the stories). In this way, the *LegendsWriter* would curate a chronicle to produce actual narrative artifacts that would be mounted in a second-order media experience. While Jacob ended up abandoning the project—he was extremely busy with *The Ice-Bound Concordance* [985, 378, 377] at the time—it contributed greatly to my growing interest in the prospect of curating simulated storyworlds.[26]

Mexica* x *World

Later in 2014, Rafael Pérez y Pérez visited UC Santa Cruz to give a presentation on his famous story generator *Mexica* [931, 936, 937, 934]. In what was my first one-on-one meeting with a visiting scholar, I showed off *World* (the subject of Chapter 7) and was delighted when Rafael responded with enthusiasm about the system and its emphasis on simulation. Before the meeting ended, we brain-

[26] Jacob is now adapting the notion of story scaffolds to his own dissertation project, *Delve* [374], which itself utilizes world generation prior to gameplay [376]. However, instead of using scaffolds to do curation, he is using them to drive world generation itself, somewhat in the style of traditional story grammars [1034].

stormed about a potential collaboration that would bridge *World* and *Mexica*.

Though the latter is an author-centric story generator that models the process of human creative writing [933]—and thus not at all a work of emergent narrative—Rafael came up with the interesting idea of having *Mexica* recount the emergent events of *World* storyworlds. We figured that doing this would require exporting *World* data in the format used by *Mexica*'s knowledgebase, which it uses to tell its stories.[27] A few weeks later, Rafael emailed me a paper on the treatment of story events as operators that modulate character emotions [932], and we planned to go from there, presumably by thinking about *World*'s emergent actions in these terms. Unfortunately, at this time I was very busy doing work on a much different topic—*videogame discovery* [1057, 1056, 1055]—and I dropped the ball on this prospective collaboration.

Looking back on this missed opportunity now, I realize that it would have represented a novel approach to curationist emergent narrative: use an existing story generator to curate storyworlds that are generated by an existing simulation engine. Such a configuration would have exemplified my argument (in Section 5.2) that the policies of interventionist systems can be appropriated as curation policies. In this case, the total operation of an interventionist system would have been appropriated for the task of automatic curation.

Generating *Prom Week* Character Diaries

My first personal foray into curationist emergent narrative was in the form of a 2015 collaboration with Chris and Matt Antoun (and several mentors), which I discussed briefly in Section 4.1.6. Our goal in this project was to automatically generate narrative summarizations of *Prom Week* [799] gameplay. Specifically,

[27] Rafael had recently carried out a collaboration by which *Mexica* was integrated with multiple other story generators in an interesting blackboard architecture called *Slant* [851].

we planned to create a modified version of the game that would generate a post-mortem after gameplay (or after each campaign) in the form of a first-person diary entry written from the perspective of a character who had been central to the events that had transpired.

This would require curation, and by now I had identified a central curationist challenge, which I called 'story recognition'.[28] Here is how we introduced that challenge in our workshop paper on this project:

> Games whose narratives emerge from simulations currently have no way of understanding the very narratives they support. Stories arise in many simulationist games only incidentally; they are remarkable streams of an otherwise overwhelming profusion of events. But while humans who play games are capable of recognizing which event sequences are storylike, the systems themselves are not. So as story generation in simulationist games works by a sort of event combinatorics, a major issue becomes *story recognition*. How does one make a system that can discern stories embedded in the morass of data that its simulation produces? [44, pp. 1–2]

As we reported in that paper, we never got to the point of automatically generating diary entries, primarily because we could not come up with an effective method for doing causal bookkeeping in *Comme il Faut* [804], as I explained in Section 4.1.6. Nevertheless, by now I was extremely interested in the prospect of curating simulated storyworlds, and this intrigue was reinforced by the failures of *Diol/Diel/Dial*—a contemporaneous project, and the subject of Chapter 8—which I identified as being rooted in a lack of curation.

Dissertation Seed

Later in 2015, with Michael Mateas and Noah Wardrip-Fruin, I wrote a paper that would become the seed for this dissertation: "Open Design Challenges for

[28] As I will explain momentarily, this was a precursor term for what I now call 'story sifting'.

Interactive Emergent Narrative" [1058], which was published in the proceedings of the International Conference on Interactive Digital Storytelling.[29] The crux of this paper was a more extensive articulation of the technical challenge of 'story recognition', to which a section was devoted. Here is an excerpt:

> There is typically no explicit narratological modeling whereby narratives in simulationist systems get composed; this would constitute a top-down approach to story generation, but simulations work bottom-up. As in other emergent narrative systems with sufficiently complex underlying simulations, stories happen in *Dwarf Fortress* only incidentally; they are remarkable event sequences among a huge boiling stew of *things happening*. While humans who play experiences like *Dwarf Fortress* are capable of recognizing which event streams are storylike, the system itself is not. We call this challenge *story recognition*: how does one make a system that can discern stories embedded in the morass of data that its simulation produces? [1058, pp. 19–20]

'Story Recognition' → 'Story Sifting'

As noted in a footnote in that paper, the term 'story recognition' was adapted from 'object recognition': "We have adapted this term from that of an analogous task in computer vision, *object recognition*, in which discrete objects are identified in image data" [1058, p. 20]. In the course of writing this dissertation, I have come to favor an alternative term, the aforementioned 'story sifting'. This progression has been due to the influence of Hayden White: whereas initially I had conceived of the task as one of recognizing stories that are already constituted (but obscured by other material), White makes it clear that stories are never fully formed in a record, but instead a record contains the *raw material* out of which an actual story may be constructed. As such, the analogy to object recognition was misguided. Instead, the task is to sift through a record to excavate the material that may be

[29] In an otherwise negative appraisal of the submission, Reviewer #2 (really) remarked, "The paper reads like the first chapter of an ambitious and much needed doctoral thesis". I hope both descriptors still apply!

used to build a story through a procedure of constructive narrativization.

Onward

By the end of 2015, my ideas about curation had been developing for nearly two years, and my framework for curationist emergent narrative, as presented in this chapter, was in the process of coevolving with my changing simulation practice. In the remaining chapters of this dissertation—those that compose Part II—I will recount this coevolution in the context of three simulation engines and three coupled second-order media experiences that, taken in total, demonstrate my increasing emphasis on curation.

Part II

How and Why

Chapter 6

James Ryan Generator

When I was growing up in the suburbs of Minneapolis in the 1990s, I had a vision for what I thought videogames should (and would) become. I yearned for a particular kind of experience that can only be had by living out an entire life, or by doing the equivalent in a computer simulation. For me, the ideal videogame experience would mean being born into a world, growing up in it, going out into it, undergoing trials and tribulations and triumphs, and then contemplating it all, with mixed pride and regret, before finally dying with the knowledge that that world would go on. I wanted to set up shop in a world, to live through seasons and make an impact, to have an authentic experience, to have a modest experience, to make real connections with people and places, to make hard decisions that I would eventually regret, to face tough challenges that I could not overcome. I wanted to be part of a generation, with all its trappings, and I wanted to see my generation percolate out of a previous one and fade into the next. I wanted the world to eventually forget me. This was the kind of narrative that enraptured me—I loved hearing stories about my ancestors and the lives that they led; I loved the generational structure of *Roots* [441], for example—and I wanted to experience these kinds of arcs on my own, through the mediation of simulation.

I figured everyone else wanted this kind of experience too, and that game designers were hard at work to make it happen. When I first encountered *SimCity* [791], I thought it had arrived, but there were no people in these cities, and though time passed in the game it did not seem to itself reflect on this changing of seasons. I read about how *Driver* [525], then upcoming, would model a real working city, but later I discovered that they were referring to a traffic system (you could not even leave your car). In *Madden NFL 99*'s [1254] newly introduced *franchise mode* [886], story time extended for the equivalent of fifteen years, but the players did not live out lives off the football field. I wanted franchise mode for the human experience. *The Sims* [792] was a revelation for me, and it almost scratched my itch, but in the end I felt that the game design was not about authentic experiences or deep connections. Eventually, I realized that other players were perfectly happy with the games that were being made, and they did not seem to need the kind of experience that I needed. Certainly, the designers of these games did not seem to hold the same vision for the medium that I did. I stopped caring about videogames.

In the final year of my bachelor's studies at the University of Minnesota ("the U" in our state's parlance), I began working with a research lab that was doing computational linguistics in the clinical domain: natural language processing on patient medical records and the results of neuropsychological tests with linguistic components. I was a linguistics major who could not program, but was qualified to do gold-standard data annotation in support of the group's mission. This mostly entailed tagging up raw clinical text so that it could be used as annotated training data for machine-learning experiments that were carried out by graduate students in the lab. It was a really interesting experience, and I even became coauthor on an academic paper [854]. The academic year passed by quickly, and when graduation was just a few months away, I had no idea what would come next for me. Upon discovering my lack of a plan, the lab's codirector Genevieve-Melton Meaux offered me a fully funded spot in a master's program in Health Informatics at the U. I entered that program and continued working with the lab.

During my first year as a master's student, the other codirector of our group, Serguei Pakhomov, began to encourage me to learn how to program. He was himself a linguist turned computational linguist, and he was keen for me to expand my skill set not just for reasons of personal growth, but because I could potentially take on more interesting lab projects. I looked into a few free online courses and tried here and there to pick up Python,[1] but it did not take. Finally, some months later, I bought a book on Python for beginners [246], and it *did* take—and then everything suddenly changed for me. I was hooked on coding, the world flipped on its head, and I knew for the first time what I wanted to do with my life.

Almost immediately, I began to apprehend the expressive power of computation. Through an early exercise in my programming book, I learned how to randomly pick an item from a list of items. This was my first authorial encounter with pseudorandomness—it was exhilarating, and it provided the seed for what would become my first actual computer program. First, I downloaded a pair of text files that I found online, one containing a few thousand masculine-coded forenames and the other a few thousand feminine-coded ones. Next, I read the files into my Python interpreter and built two lists, each containing all the names included in one of the files. Finally, I wrote a short program that randomly picked a name from one of the two lists and another name from the masculine-coded list, and stuck them together. Because many masculine forenames are also used as

[1] At this time, in 2011, Python had recently emerged as the preferred programming language among computational linguistics. Were it 2001, I would have proceeded to Perl, and perhaps thence to *Perl golf* [848].

surnames, the resulting creation read like a person's full name.[2]

Even though it was only a name, the result was somehow evocative: it felt like I had created a tiny abstract person who lived in my computer. Names denote persons, and I—having a mind and seeing a name—could not help but conjure up ideas about the people behind these generated names. As a child, my grandmother took me along each Memorial Day to place flowers at the gravestones of our ancestors who are buried across a series of cemeteries in Minneapolis and St. Paul.[3] To me, the gravestones, with their names and dates, constituted a vast narrative possibility space that was overwhelming to walk through. I wondered what these people had been like, and what this world had been like for them. *What kind of person would have that name? What would it be like to have that name?* I was now undergoing this kind of experience again, but through the mediation of computation. And I was not just experiencer—I was author.

My next experiment emerged out of the realization that my own name, both first and last, appeared in the corpus of masculine names. I made a variation on the program that generated names over and over again until it generated my name, at which point it would print both my name (as proof) and the number of iterations that it took; a reconstruction of this program is shown in Figure 6.1. I recall it typically taking around nine million iterations—though it took only a few seconds to run, which was flabbergasting—and each time the result was strangely haunting: a little computer person with the same name as me. Or was it actually me—was this a kind of self-portrait? This was the first computer program that I named. I called it *James Ryan Generator*.

Over the next few months, I became more and more immersed in the aesthetic possibilities of computation, and my name generator evolved into a more complex

[2]Of course, not all cultures use this kind of naming practice.

[3]I am a sixth-generation Minneapolitan on my father's side and a sixth-generation St. Paulite on my mother's side.

```
import random

NAMES = open('names.txt').readlines()
NAMES = [n.strip('\n') for n in NAMES]

i = 0
name = None
while name != 'James Ryan':
    i += 1
    first_name = random.choice(NAMES)
    last_name = random.choice(NAMES)
    name = first_name + ' ' + last_name
print name
print i
```

Figure 6.1: A reconstruction of *James Ryan Generator* (2012), my first named computer program. After reading in a corpus of masculine forenames—one that includes both 'James' and 'Ryan'—the program randomly produces synthetic full names until it regenerates mine, at which point it prints out the name (as proof) along with the number of iterations that it took for it to be regenerated. I was mesmerized by this simple name generator, and it led me down the rabbit hole of simulation (and procedural generation, more broadly). Within a few months, I had expanded it into *World*, the subject of Chapter 7.

program that I called *Lineage*. In this system, the generated names were treated as tiny characters who existed in an abstract virtual world where time passed one year at a time. Each year, the characters grew older, and by an act of random selection new characters could be born from existing ones (with a surname being inherited patronymically). Eventually, the characters would pass away, leaving their virtual world, and thereby my computer's memory and all of existence: *for dust you are and to dust you will return.* The ephemerality was powerful.

Now there was not merely an abstract person, but a series of them who were interacting across the passage of time—I had built my first *simulation*. Moreover, while the generated names had been evocative to me, like the ones I saw in those Twin Cities cemeteries, the unraveling of tiny little lives across *Lineage*'s simulated epochs had a distinctly narrative quality. It was not just names and dates, but a series of events that unfolded through the passing of time. If this simulative unraveling was a kind of story, or something that contained a story, it was a story with no author. Even though I had crafted the program and fully understood how it worked, I could not claim authorship over the stories that I was recognizing in its outputs. I had embarked on a project in emergent narrative.

My nascent computational practice had formulated tiny replications of me, and now it was working to reformulate the actual me. The world looked different than it had before—I took on a computationalist perspective. Systems appeared everywhere, and many of the media experiences that I had appreciated seemed to lack richness. I began thinking again about the kind of videogame experiences that I had yearned for my whole life, and I realized that I had been inadvertently working toward that vision in my first few months of computer programming. After ten years of ambivalence, I became interested in games again, and I was excited to discover that a new category of narratively focused independent projects had emerged in the interim. Most of these were not simulationist, but they were still intriguing to me. One project, however, was clearly in the same spirit of my own burgeoning practice: *Dwarf Fortress* [17], the simulationist tour de force that has appeared throughout this document. It seemed like there were likeminded creators out there who were working to materialize the kinds of narrative possibilities that I had spent my life anticipating. I was invigorated and decided that this was what I wanted to do with myself.

In the summer of 2013, one year after learning to code, I arrived at the University of California, Santa Cruz, where I began the doctoral stint that is now culminating in this dissertation. More specifically, I became a member of the Expressive Intelligence Studio, a cultural and technical research lab that is dedicated to exploring new possibilities for art and entertainment that are enabled by techniques from artificial intelligence.[4] It was in this context that I discovered the application area of artificial intelligence called *story generation*, in which computer programs are made to tell stories. This was quite an intriguing idea for me, since I had never thought about computational narrative outside of the context of the interactive experience of a videogame.

As my horizon expanded in this way, I began to reconsider my vision for the ultimate videogame. I was still interested in being a part of a dynamic virtual world that would be full of meaning and that would change over time, but I realized that I was also just fundamentally interested in the idea of that kind of world and the little computer people who might live in it. This predisposition had led to me being more interested in non-player characters (NPCs) than players, and in story generation I found the purest expression of this philosophy: virtual worlds with no players. By now, my little simulation had evolved into something larger that I called *World*—the subject of Chapter 7—and while I was finding ways to put myself and others into its simulated storyworlds, I was also interested in letting it run without human intervention. I had intended for the framework to serve as the backbone of a simulationist videogame, but now I was beginning to also view it as a potential contribution to the area of noninteractive story generation. This ecumenical view on the issue of interactivity is an idiosyncrasy of mine that has

[4]For my first year, I was also a member of the Natural Language and Dialogue Systems lab, which is led by Marilyn Walker. My research agenda was initially limited to the exploration of new techniques for dialogue generation in games, but eventually it grew to encompass my simulation practice and a few other areas.

worked to structure both my creative practice and my interpretative work, and that is why this dissertation has not been partial to interactive examples. I do not really care whether a work is interactive.

As my simulation practice and conceptual vocabulary coevolved here at UC Santa Cruz, I began to see my work as fundamentally being a contribution to emergent narrative, both interactive (in the tradition of simulationist videogames) and noninteractive (in the tradition of simulationist story generation). Funnily enough, though, I soon realized that I was a protégé of a fairly vehement opponent of emergent narrative in my field: my coadvisor, Michael Mateas, tends toward the *interventionist* mode of computational narrative (to use the term introduced in Chapter 5). His approach to interactive storytelling—as exemplified in his collaboration with Andrew Stern, *Façade* [775]—is one of dramatic remediation: a system, called the *drama manager*, administers the behavior of autonomous characters so that the experience follows a pleasing dramatic arc [776].[5] A more *emergentist* alternative, where a designer trusts that the outcomes generated by a bottom-up simulation will be pleasing enough, is anathema to him: *it's just one damn thing after another*. While I think he is less opinionated on the matter, my other coadvisor, Noah Wardrip-Fruin, has written on potentially inherent limitations of the pure simulationist approach, as characterized by his critique of *Tale-Spin* [1310, 1311], which I discussed in Chapter 4.

But while I deeply considered the virtues of interventionism, I could not shake my allergy to it. For me, a simulated storyworld is a beautiful thing: it is pure and real—literally real—and the tiny abstract characters who live there are having real and authentic experiences, albeit tiny and abstract ones. When interventions are

[5] To be clear, I absolutely love *Façade*—it changed my life and remains one of my favorite works of computational media—but in my own practice I am beholden to an emergentist impulse, as I will explain momentarily.

made, the storyworld is polluted, and the result is less intriguing to me.[6] In considering this, I eventually realized that while the near consensus in academia is that emergent narrative is a dead end, it seems to somehow also constitute arguably the most successful approach to procedural narrative in the public consciousness, through its expression in videogames like *The Sims* and *Dwarf Fortress*. How could this be? As I stated in the previous chapter, I believe the answer lies in the tagline *overgenerate and curate*: emergent narrative works in projects whose simulated storyworlds generate an overabundance of narrative material, which some entity (usually a human player) then curates to construct a story. This realization came to me gradually over the course of building three simulation engines and an array of second-order media experiences that are driven by them. In this sense, my framework for curationist emergent narrative—the nucleus of this dissertation—coevolved with my simulation and media practice. This is the story that I will tell in the following chapters.

At its heart, this dissertation is about an idea and a series of artifacts that I have developed over the last six years, either in collaboration or independently. For me, this work has always been in pursuit of the vision that I articulated at the beginning of this chapter. I want to encounter stories that recount simulated worlds, and I want to have authentic experiences in those worlds. This is a vision that I have held my entire life, and it continues to drive me today. It has worked to constitute not only my creative and intellectual life, but in a sense my entire life—it is, in its own way, a James Ryan generator.

end intro

[6]As I have noted several times by now, here I am using the charged language of an art manifesto. To be clear, this is not meant as a technical or even an intellectual argument, but rather an artistic one. This is how I feel.

Chapter 7

World

My name generator had evolved. I was hooked as soon as I saw the first synthetic names—it felt like behind each one was a tiny little computer person, and they were appearing one after the other. But while they emerged out of the same procedure, on the same machine, each was partitioned into its own tiny universe constituted entirely by itself—by one pregnant name. I could generate multiple names in one run, and maybe then there was a stronger sense of cohabitation, but what I really wanted was for these little people to interact. I still did not know how to do much with code, but I realized that I could do something interesting by simply reapplying the procedure that had enabled the name generator in the first place: picking randomly from a list of items.

In an evolved form, the program now worked like this: generate a series of names, compile these into a list of synthetic names (people), and then randomly select a pair of names *from that list*. Now I had all the tiny people living in the same universe—the list containing all the generated names—and I had a means of deciding who would interact with whom. But now I had to figure out how to model an interesting interaction between the randomly selected pair, which was a significant challenge given my very limited coding skills.

I have always been interested in genealogy, and that predilection would drive the evolution of this system beginning with my initial modeling of character interaction. As established, the program to this point was almost entirely based on the single operation of selecting randomly from a list. My next move was to apply this *again* to create new computer people as the offspring of existing ones. This was done by taking the pair who were randomly selected to interact, checking if procreation would be possible between them, and, if it was, creating a new computer person. To do the middle step, the system needed a means of representing the sex of a computer person, and so this became the first modeled character attribute beyond one's name. As noted in Chapter 6, the names corpus that enabled this whole enterprise was composed of two files, one containing male-coded names and the other female-coded ones. Instead of selecting a first name from the entire corpus, the system now randomly determined the sex of the character first, and then picked a name from the corresponding gender-coded list. The system also kept track of the sex that was randomly selected, which meant the little computer people were now represented as tuples comprising one's sex and name. To create a new person, the system carried out this same procedure, but instead of randomly selecting a surname, it operationalized a patronymic scheme by which the father's surname would be inherited. This required further evolution of the character representation, since the monolithic name now had to be broken into its constituent parts, so that the program could specifically identify surnames.

In this way, my system kept evolving into an increasingly elaborate hack—an embryonic simulation engine, and then eventually a full-fledged one—and along the way I learned how to program. One critical addition was the modeling of time, which was introduced to overcome the otherwise inevitable situation of all the people living indefinitely as immortals. This took the form of a simulation timestep, treated as a full year, during which new people might be born. As part of this reformulation, the system kept track of everyone's age, and with each passing year there was a chance of death. At this stage, I called the program *Lineage*, and viewed it as a genealogy simulation. After introducing record keeping about who gave birth to whom (the character representation now included one's parents and children), I could explore the generated family trees by chaining backward or forward along the filiation links. It was fun to see which lineages would persist according to the happenstance of the random number generator. By changing very little code, I could also explore alternative naming practices—a matronymic scheme for surnames, or babies being named after family members or other people in the simulated world.

As for that world, the next critical hack was to introduce a modeling of space, by which I intended to inject more intrigue into the genealogy simulation. Whereas the world had been totally abstract as far as its physical representation, now it became an open sea scattered with islands. This was achieved by representing the world as a coordinate grid for which certain coordinates were randomly selected to have islands. To move about the world, the tiny computer people could build tiny computer ships, which allowed exploration into adjacent world coordinates. If an island was encountered, the seafarers would disembark and carry on with their abstract little lives on the new island. This richer modeling introduced interesting nonlinear dynamics into the genealogy simulation, and it elicited new ideas about how I could make it even richer. From here, my programming skill began to ramp up more quickly, and in turn the simulation detail did too. I now called this evolved form of the system *World*.

Within a year of creating the name generator, I had been accepted to UC Santa Cruz, where I would commence on a research agenda dedicated specifically to the intersection of natural language and videogames. Thus, my early official projects were not in the area of simulation or even emergent narrative, but rather expressive natural language processing [1051, 1052, 1053, 715], though I continued to develop *World* as a side project.[1] In the early months of 2014, I took Noah Wardrip-Fruin's class *Playable Media* [1313], which requires the students to distribute a significant project to the rest of the class for critique. For this, I chose to build a text adventure that would harness *World* as a *simulation engine*.

By 'simulation engine', I mean a framework that is itself a work of media, but one that is primarily intended to fuel *second-order media experiences*. In my configuration, *World* would drive a text adventure called *Islanders*, and so the underlying simulation would not be experienced directly, but rather through the higher-order mediation of a designed interactive experience. This pattern is a calling card of my simulation practice, and this chapter and the following ones will be structured accordingly to introduce, one after the other, my three simulation engines along with three second-order media experiences that are driven by those engines. It took me a while to realize the power of this decoupling—namely that a single simulation framework, viewed as an engine, may drive multiple experiences that take alternative approaches to the process of curating its simulated storyworlds. As a result of this, the few people that are aware of this project know it as a monolithic thing named either 'World' or 'Islanders'. Again, to be clear, *World* is the simulation engine and *Islanders* is a text adventure built atop it.

While *Islanders* was not particularly successful, it was through conversations about the project with Noah and also Michael Mateas (by then they were my sole coadvisors) that I learned that I could decide my own dissertation topic, or more specifically, that it could change. For some reason I had viewed my

[1] At this time, I was also a member of Marilyn Walker's Natural Language and Dialogue Systems lab at UC Santa Cruz.

statement of purpose (in which I proposed to do videogame dialogue generation) as a kind of binding contract, but after a friendly comment from Michael—"I hear your *Islanders* project is transmogrifying into a dissertation topic!"—I decided to incorporate my simulation practice into my actual research agenda.[2] The result was initially a kind of loose exploration of emergent narrative surrounding conversationally competent characters in a massive simulated storyworld, which is especially apparent in the publications on *Talk of the Town* [1061, 1059, 1045, 1050, 1220, 1044], the subject of Chapter 9. By the end of 2014, I had to decided to abandon *World* to do a technical and intellectual reboot in the form of *Talk of the Town*.

As my simulation practice was developing, my thinking about emergent narrative coevolved in turn, and the latter would eventually solidify into the framework at the heart of this dissertation: curationist emergent narrative. This coevolution pertained especially to the idea I called *story recognition*, which was introduced in two 2015 papers [44, 1058] as the technical challenge of building a simulation that could recognize its own interesting emergent stories.[3] While *Islanders* is the *World*-driven media experience whose development was most extensive, I have chosen to use another project, *Diol/Diel/Dial*, as the case study following this chapter. This is because that project, a hypertext encyclopedia generated for a *World* world, was actually more critical in the development of my increasing technical and intellectual emphasis on curation in emergent narrative. Nonetheless, because the development of *World* was so tangled in the development of *Islanders*, I will inevitably tell the story of the latter in the course of explaining the former.

While it evolved gradually out of my name generator, and the intermediate

[2]When prospective graduate students in the new department of computational media visit UC Santa Cruz, I attempt to illustrate the wonder of this place by explaining how all my weird side projects (that I would have done anyway) somehow became official research.

[3]I now prefer the term 'story sifting', as I explained in Chapter 5.

system *Lineage*, this chapter is about *World* at the final stage of its development, constituted in the code base that was used to generate *Diol/Diel/Dial*. Like any computer program, my simulation engines have evolved over time, but in the following chapters I describe each system as it was structured in driving the case study of the coupled chapter.

My intention in these chapters is multifarious, in that I seek to: reflect on my past, provide an autoethnography of my simulationist media practice, report technical contributions, report intellectual contributions, further illustrate the idea of curationist emergent narrative, demonstrate the feasibility (and challenges) of that framework in terms of completed projects, articulate a poetics of that approach learned from successes and failures, and to sketch a series of open technical and design challenges. The larger story of this entire retrospective, spanning from *James Ryan Generator* (2012) to *Sheldon County* (2018–), is one of an increasing emphasis on practices of curation by which the stories that emerge from massive world simulations may be recognized and then mounted in full-fledged media experiences. These are the reflections of a *simulationist-cum-curationist*.

This chapter and the subsequent ones will take the following structure. First, I will recount the impetus for developing the chapter's subject artifact (simulation engine or media experience), as I have just done for *World*. Next, I will explain how the artifact works. Finally, I will discuss the artifact. In the chapters devoted to case studies of media experiences, I will also provide a self-critique that is driven by the intellectual framework that I developed in Part I of the dissertation to appraise all of emergent narrative (pleasure, pain, curation). As such, I will be using these weapons of appraisal against myself. Critically, the resulting series of critiques will demonstrate an evolution toward artifacts that maintain the pleasure of emergent narrative while simultaneously alleviating the pain, through an increasing utilization of curation. This evolution is in fact a case of coevolution: my ideas about emergent narrative—which eventually solidified into the curationist framework—coevolved along with my simulation and media practice. As such, these chapters tell a self-referential story of how this dissertation itself came to be; if you like strange loops, just wait.

Before moving on, I should say more about my approach to evaluation in this chapter (and the subsequent ones). I believe that computational systems that are intended to support media experiences can only be appraised through actual implemented experiences that are built atop them; others have articulated this before me [762, 1200, 505, 610, 1316]. Because the purpose of *World* is to drive second-order media experiences, I am going to hold off on appraising it in that regard until the following chapter, in which I will critique it through the lens of *Diol/Diel/Dial* (an actual implemented media experience built atop it). In lieu of such appraisal, in this chapter I will instead situate *World* against related technical work, both in a broad sense of its approach to simulation and also in terms of some of its specific subsystems, particularly the one that models character language phenomena. (This alternating structure will recur in the following chapters, as well.) For clarity, I discuss this related work in a distributed manner, throughout the chapter, as the pertinent concepts arise in my discussion.

As a final note, *World* has been reported in one paper that was dedicated exclusively to its language modeling [1037], but otherwise this chapter tells its story for the first time.

7.1 Preamble: *Islanders*

Islanders (2014) is a text adventure driven by the *World* simulation engine. Since the development of both projects was deeply intertwined—*intertwingled*, even [876, p. DM31]—I will first briefly outline the former before proceeding to describe, in the rest of the chapter, how the latter works.

Living a Life

Prior to *Islanders* gameplay, a *World* world is generated and then simulated for 100 years, at which point the player is randomly assigned to the womb of a pregnant woman in the storyworld; the next year, the player character is born. In this way, that player character is in fact descended from non-player characters (NPCs), and until adulthood major decisions (such as moving between settlements or embarking on ship voyages) are made by the character's NPC parents. At any time after age five, the player may learn about nearby people (by navigating a menu containing this information) and visit local cemeteries (to view the gravestones for characters who lived and died earlier in the simulation). Upon reaching adulthood, the player may have her character engage in any of the activities that NPCs carry out (these are discussed below in Section 7.3).[4] As such, the set of *player affordances* [1241] is essentially the set of actions NPCs may take in the simulation. Death strikes at some point (from any of the potential causes that listed below) and is permanent, following the *roguelike* style [213, 174, 917].

Afterlife Affordances

The game is text-only during the player character's life, but upon her death an interactive world map and hypertext world encyclopedia are displayed. Eventually, I published one of these generated world encyclopedias as a standalone project—this is *Diol/Diel/Dial*, the subject of Chapter 8. Additionally, the player may continue to progress the simulation to see what becomes of people and places

[4] A few of these, namely propagating knowledge, are not available (mainly due to dialogue interactions not being implemented).

of interest (such as offspring or a founded settlement), which will cause the map and encyclopedia to update.[5] The figures below that depict segments from *World* worlds are screenshots of such interactive maps.

Targeting Regret

With *Islanders*, I aimed to develop a game that was extremely open-ended while still yielding narrative potential, and one that approximated my lifelong vision, recounted in Chapter 6, of having the experience of living out an entire life in a tiny world. Many players expressed a feeling of regret at having wasted their little lives, which I came to view as the primary aesthetic achievement of the project.[6] As I mentioned in the previous chapter, I would like to see more videogames that elicit regret—worlds in which you cannot be fulfilled.[7] Generally, the *Islanders* project was not successful, and this was likely due in large part to the major deficits in the underlying simulation that I discuss in Section 8.4.

7.2 Modeling

Each *World* storyworld is modeled as a large sea scattered with island archipelagoes that are populated by characters, who construct houses and ships and form settlements. In this section, I will give a basic overview of the modeling of characters and the various entities that are included in the simulation. As a brief note,

[5]The notion of simulating ahead past the time of gameplay would later serve as the basis for the *Bad News* epilogue scene, discussed in Section **??**.

[6]One humorous example was the experience of Aaron Reed, who decided to become a great explorer but died at sea on his first voyage. Upon his character's death, the world map was revealed, and he discovered that he had been born into a world with only one island; Figure 7.2 depicts such a world. In another life, Sarah Harmon compulsively built ship after ship and then spent her final years on a sabbatical, asking herself why.

[7]In this sense, *Islanders* was an attempt at *memento mori* [865], which was Jason Rohrer's aim in developing *Passage* (2007) [1017], as he has stated [1018]. Earlier this year, Rohrer released a game called *One Hour One Life* [1019], which also explores genealogical phenomena.

in developing this setting, I was loosely inspired by the settlement of the south Pacific [419], and in particular the culture and practice of *Polynesian navigation* [1143]. Moreover, though, as I mentioned above, my core aim in *World* was to support emergent genealogical intrigue.

7.2.1 Time

Time is modeled very coarsely in *World*, with *timesteps* that correspond to one year of storyworld time. Within a timestep, there is no explicit modeling of finer increments of time, but there is an implicit temporal ordering in that the procedures that generate events are grouped according to event type. For example, character births for a given year occur all at once, then later in the simulation of that year character deaths all occur at once, and further on gravestones weather all at once, and so forth. At the beginning of world generation, the year is set to `0`, though *primordial characters* who are already alive at that time will have retroactively modeled birth years that are represented in the calendar using negative integers (e.g., a birth year of `-22`). Time proceeds one year at a time until world generation terminates (once a configurable termination date is reached). While this timestep size is probably too coarse, as Section 8.4 discusses, this engine was designed to generate stories that may take place over decades or centuries (following my vision for procedural narrative that was outlined in Chapter 6).

7.2.2 Characters

In my simulation practice, characters are central, and this has been the case since the name generator that preceded *World*. In this section, I will describe the various modeled components of a *World* character.

Name

As I explained at the beginning of this chapter, *World* evolved out of the name generator that underpinned *James Ryan Generator*, and so character *names* are in a sense the bedrock of the technological substrate underpinning the engine. In the system, character forenames come from the same corpus that drove the name generator, which comprises several thousand female-coded names and several thousand male-coded names. Surnames are drawn from a corpus of common examples of European origin.[8] When a new character is born, the parents select a first name and a middle name and then use the father's surname. The forenames are determined by randomly selecting from the corpus or by probabilistically selecting a family member to name the child after.[9] Children who inherit the exact name of a parent will receive a numeral suffix indicating this. Additionally, if a child is named for someone else, the system records this information and then chains backward to determine for whom the character was *ultimately* named (in the case that the namesake was in turn named for someone else, and so forth). The modeling and tracking of familial naming practices reflects my core aim in developing *World*: produce emergent genealogical intrigue.

Personality

World uses the famous *five-factor model of personality* [810, 1337], which is also referred to as the *Big Five* model (due to the coinage of Lewis Goldberg [401]) or the 'OCEAN' typology (e.g., [302, 277]). This model has a long history in psychology that begins with pioneering work by Sir Francis Galton [1151, p. 30]

[8]I clearly made no effort to explicitly model the Polynesian setting that influenced me. Generally, *World* is an aesthetic hodgepodge of miscellaneous content and sometimes incongruous modeled phenomena. This reflects its development, which was carried out by bricolage: the codebase evolved haphazardly as I learned new programming tricks and made use of a peculiar assemblage of materials and corpora at my disposal.

[9]Here, probabilities are associated with the various familial relations.

and traces through a series of refinements carried out by a number of researchers across the twentieth century [547]. While various alternative names have been attributed to the model's five dimensions [264, p. 423], they are conventionally known as *openness to experience*, *conscientiousness*, *extroversion*, *agreeableness*, and *neuroticism* (hence the 'OCEAN' acronym). In the model, each trait is treated as a dimension of personality, with poles on each end that correspond to opposite extremes in terms of how the trait may express across individuals.

In *World*, each trait is modeled as a floating-point value between `-1.0` and `1.0`. When a new character is born, her personality is determined in this way: for each trait, randomly select which parent she will take after, mutate that parent's value for that trait, and set the mutated value as the child's value for the trait. To mutate a trait value, the system treats the parent's value as a mean and then generates a pseudorandom number from a normal distribution that is centered on that mean, with a standard deviation corresponding to the degree of heritability for the trait at hand (according to a study by Kerry Jang and others [538]). For example, openness to experience has been found to be more heritable than agreeableness, so in turn the mutation term (standard deviation) for the former is less than the one used for the latter (which means *World* characters tend to take after their parents' openness trait values more closely). In the case of a character who already exists at the beginning of world generation, trait values are pseudorandomly generated from trait-specific normal distributions that use means and standard deviations identified in a study by David P. Schmitt and others [1116]. Character personalities do not change over time—this may or may not be realistic, as some researchers have found that Big Five profiles change over time [437, 1015, 1190], while others have identified a general stability [1182, 195].[10]

[10]The matter essentially boils down to *nature versus nature* [947], as Sanjay Srivastava and collaborators illustrate in their paper on the topic [1190].

In developing *World*, I was attracted to the Big Five model because of the prevalence of studies that use it to explain human phenomena in terms of personality variation, which afforded easy *operationalization* [63]: to model something new, I could quickly look up accounts of how human personality variation drives the phenomenon in the real world. By this approach, such descriptions of human behavior can be inverted by treating a correlative relationship (between personality trait values and a behavior) as a causal one—that is, if a study describes a certain behavior (e.g., lying) as being more prevalent among individuals with certain traits, then in my simulation I make individuals with those traits more likely to engage in such behaviors. Since simulation tends to loosely model real-world phenomena, operationalization represents a fundamental approach to the craft, as *Dwarf Fortress* [17] cocreator Tarn Adams has noted:

> It's helpful to base your simulation on reality. If you have a real-world analog in mind, you can correct defects using a broader understanding of the fundamentals. [...] We know the real world works, so if you fall back on reality, you can usually get the simulation to work as well, given adequate memory and CPU time. [15, p. 520][11]

Indeed, Adams's polymathic operationalization practices undergird the famous joke that his game's *design document* [363] is Wikipedia, as Adams has acknowledged in interviews:

> I mean take the geology for example. [...] I wanted the rocks to be, to coincide with the reality of it. So I went basically—I mean people have joked that Wikipedia is a design document for Dwarf Fortress or whatever. Because I just went through it. [...] So, yeah, but lots of research. And that goes back to one of those skills I was talking about when you're doing a simulation is being able to take something like a Wikipedia article, which is just a summary of purported facts. And you can take that, find out what the core elements are, figure out what rules govern them, figure out how you're gonna turn that into a

[11] I love the phrasing of 'we know the real world works', which to me evokes the roboticist Rodney Brooks's famous quip, "the world is its own best model" [144, p. 5].

structure in a computer game and algorithms in the computer game [1083, pp. 24–25]

As I argued in Section 4.1.5, simulation crafting takes a lot of authoring, and the results will always fall short of ideal modeling. In my experience, the only way to actually build intriguing generative storyworlds is to model as much phenomena as possible in the shortest amount of time. To achieve this, it is key to make reasonable decisions quickly, and to conceive of a simulation engine as an endless work in progress. If the simulationist deliberates too much, her engine will probably never develop into something whose modeling is rich enough to produce interesting storyworlds. There is simply too much authorial burden to spend much time planning, and so good simulationists are fast satisficers. With regard to this notion, consider *Dwarf Fortress*, which is almost certainly the richest simulation ever crafted: Tarn Adams has been working on it continually for the last fifteen years, and he is only halfway through his planned development period. Even though he planned from the beginning to spend decades on the project, Adams has always been keenly aware that there is no time for major deliberation:

> I took my current 30 personality facets from the NEO PI-R test, for instance, without having much knowledge of the theory. If it fits in to the game, makes sense to me and provides enough diversity to be reasonably resilient to future changes, I can use it. [826, n.p][12]

Another major appeal of the Big Five model for me was that it was also being harnessed for personality-driven expressive natural language in the work of Marilyn Walker's Natural Language and Dialogue Systems lab [872, 1301, 1303]: namely François Mairesse's dissertation system *Personage* [728, 729, 727, 730] and

[12] Adams has since revised his personality model to include several more traits. His additions reflect an increasing tendency away from realism and toward models that better support the emergence of dramatic material. This shift represents the crossing of an idealogical divide pertaining to the meaning of 'believability' in character modeling, which I introduce momentarily. As I discuss in Section 11.1.2, I have also moved in this direction with regard to my simulation practice, and in fact I am now using Adams's refined *Dwarf Fortress* personality model.

the ambitious prototype *Spyfeet* [987, 986], the latter of which was a collaboration with members of the Expressive Intelligence Studio. During this period of my simulation practice, I was on my way to UC Santa Cruz to join both of these labs, where I anticipated working with the five-factor model—as such, it made sense to experiment with it in *World*. When I got there, however, I was surprised to discover that my new coadvisor Michael Mateas was not a fan of the model. This was my first encounter with Michael's ideology surrounding *character believability*, and it foregrounded a fundamental tension in computational media that I had not been aware of prior to arriving at Santa Cruz.

Though it is widely agreed that believability is a quality of fictional characters who make it easy for the audience to suspend its disbelief, some practitioners assume that such quality is rooted in a kind of realism, while others have contended that it is in fact orthogonal to realism.[13] As Michael explains in his 1997 review of interactive drama, the term 'believability' entered computational media under the influence of the *character arts* [1247, 551]:

> *Believable* is a term coming from the character arts. A believable character is one who seems lifelike, whose actions make sense, who allows you to suspend disbelief. This is not the same thing as realism. For example, Bugs Bunny is a believable character, but not a realistic character. [758, pp. 7–8]

Michael's stance is paradigmatic of the tradition of the Oz Project [84, 83], the research lab (and computational-media studio) with which he was affiliated as a graduate student at Carnegie Mellon University. In his capacity as figurehead of that project—the "Wizard of Oz"—Joe Bates took the emerging influence of the character arts on computer animation [649] and extrapolated it to the broader domain of autonomous characters. This was the basis for Bates's coinage 'believable agents' [85, p. 123], but the subtle notion of 'lifelikeness' has often been taken to

[13] For a nice overview, see this paper by Josh Tanenbaum and Jim Bizzocchi [1240].

mean 'realism'. In any event, in this period of my simulation practice, I strove for realism, but over time—through operating in the tradition of the Expressive Intelligence Studio, which itself operates in the tradition of the Oz Project—I would come to align more with the perspective that aims for a more stylized believability. In Section 11.1.2, I tell the story of my transformation.

Incidentally, character personality does not do much in *World*, since it was added in toward the end of the project. For the most part, it drives friendship formation and aspects of the language simulation, as I will explain below.

Disposition

In addition to a five-factor personality model, *World* characters have a *disposition* comprising traits called *vigorous*, *violent*, and *depressive*. While character personalities, a late addition to the simulation, do not do much to drive character behavior, dispositions do, as I will explain below. In this simple model, violence and depression are binary attributes (either present in a character or not), and vigor is a floating-point value that decreases with a character's age. While the introduction of the five-factor model represented a shift toward realism that would be most pronounced in *Talk of the Town*, the subject of Chapter 9, initially I had gravitated toward a more dramatic model, as the disposition traits show. In *Hennepin*, the subject of Chapter 11, I shift again toward the dramatic, but in doing so I attempt to strike a balance.

Language

World characters may acquire *languages* as they go about the simulated storyworld, including native languages (during adolescence) and non-native ones (after adolescence). For each language that a character speaks, she maintains a proficiency level (represented as a floating-pointer number between `0.0` and `1.0`) and instantiates an *idiolect* that represents the unique expression of that language in her speech. As I explain in Section 7.3.16, which reports extensively on this subsystem, languages are represented abstractly using a vectorial scheme.

Knowledge

A character's accumulated *knowledge* of the storyworld is structured as a collection of beliefs about the last known whereabouts of other characters. Section 7.3.14 is dedicated to the simulation of character knowledge phenomena in *World*, so I will hold off on explaining more until then.

Miscellaneous Attributes

World characters are also modeled according to a few additional attributes beyond a name and personality. These include *sex*, *sexuality*, *fertility*, and an extensive array of attributes that represent the character's *family tree*; a character's sexuality is composed according to whether she is probabilistically determined to be attracted to males and/or females. Along with the naming simulation, these attributes reflect my core aim of supporting genealogical intrigue, since these are some of the basic mechanisms that drive the phenomena that are captured in genealogical records. Additionally, a character *memory* attribute is modeled as a floating-point value that decreases over time, making forgetting (about which more soon) more likely. Finally, the system tracks character-specific data pertaining to a number of concerns including social relationships, romantic history, residences, homes constructed, ships constructed, settlements founded, ship voyages, islands discovered, crimes committed, and more.

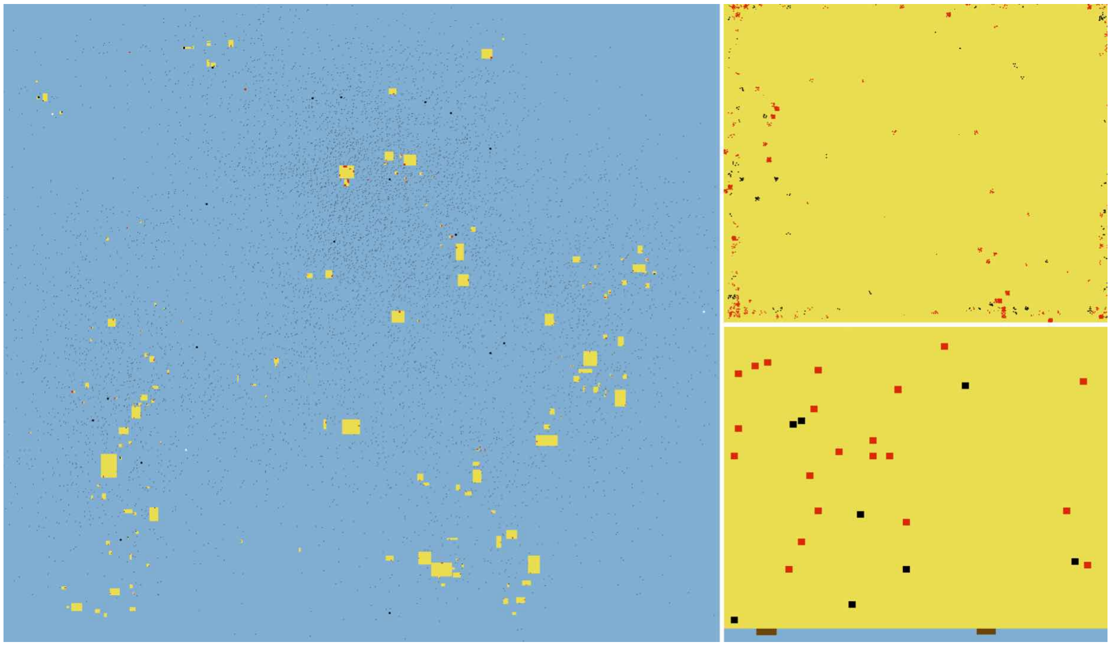


Figure 7.1: A *World* world, along with a close-up of an island and a port settlement (organized clockwise). In the world map, yellow rectangles represent islands, red rectangles are active settlements, black rectangles are abandoned settlements, black dots are ships, and speckles represent earlier ship positions. In the island and settlement close-ups, inhabited houses are red and empty ones are black; ships are shown as brown rectangles in the port view.

7.2.3 Other Entities

In addition to characters, *World* models the world, the ocean, islands, settlements, houses, ships, artifacts, languages, and whales. In this section, I will describe how each of these entities are modeled in the system.

Worlds

A *World world* is modeled as a square grid whose coordinate system spans 10,001 units on each side. I conceived of these units as corresponding to kilometers, which makes each storyworld roughly the size of the Atlantic Ocean. Each world is scattered with islands, but constituted primarily in a single massive ocean, as

Figure 7.1 illustrates.[14]

Oceans

The *ocean* of a world is constituted in the portion of the world's coordinate system that is not land, and implicitly it extends further, too: islands are never placed outside the world boundaries, but ships may cross the threshold and continue to traverse by sea into the empty exterior.[15] In this sense, each storyworld's ocean, and thus each storyworld, has no explicit boundaries.

Islands

A world's *islands* are abstractly modeled as rectangles that are defined by the coordinates of their southwest and northeast corners. At the beginning of world generation, the system determines how many islands there will be (n) by generating a pseudorandom number from a normal distribution ($m = 49$, $sd = 20$); if this number is less than or equal to zero, the procedure is repeated. In rare cases, a *World* world may have only one island, as Figure 7.2 illustrates. Next, these n islands are distributed across the world by randomly selecting center coordinates. As a new island is being created, its size is determined by determining a height and width, again by generating from a normal distribution ($m = 20$, $sd = 44$). Additionally, a new island may anchor an archipelago. If a probabilistic

[14] I spent the summer of 2014 developing systems for terrain generation, with the intention of using it as the basis for a new version of *World* that would make use of such detailed modeling. As fate would have it, my next simulation engine was *Talk of the Town*, for which the technology for terrain generation was overkill. I am now working to finally integrate this work back into my practice in the form of terrain generation for *Hennepin*, which supports interesting dynamics surrounding town development and character actions.

[15] Because there is no hope for finding land outside the boundaries, crossing the threshold spells doom. Once everyone on board perishes, however, the vessel may voyage onward as an uninhabited ghost ship, blowing in the wind. I have seen ghost ships that emergently traveled many thousands of units past the world boundary. Conceptualizing the coordinate system as continuing to extend beyond the expanse of my computer screen, a few stray ships traveled nearly to the walls of my apartment.

threshold is eclipsed by a generated number, between one and six nearby islands are generated in the vicinity of the anchor. Beyond its size, an island has no topography, but the system reasons about which coordinates are on which island coasts, and corners have the special property of being on two coasts.

Each island is unnamed until it is encountered by characters, at which point a probabilistic procedure is enacted to select a name. By this procedure, an island may be named in homage to its discovery ship's port of origin—e.g., 'New Death Isle' for 'Death Isle'—or in honor of the ship's captain or that character's loved one. Alternatively, a name may be randomly selected from a corpus containing the names of American municipalities; here, the name 'Minneapolis' is fifteen times more likely to be selected.[16]

Wilds

Just as a world's ocean is defined as all of its area that is not land, an island's *wilderness* is constituted in all of its area that is not the site of any settlement (about which more next). Because houses never disappear, as I explain shortly, settlements that become abandoned likewise remain forever, which means the wilderness of an island will strictly shrink over the time (or remain the same size, if that island is never discovered). A wilderness is always named descriptively—for instance, 'Wilderness of Death Isle'. As I explain below, characters may venture out into the wilderness of an island, which is one of the ways in which new settlements may be founded.

[16]Diol, the storyworld recounted in *Diol/Diel/Dial*, features several places called Minneapolis, including a curious island discovered by a small crew of six characters who would each die there, leaving it uninhabited: `https://users.soe.ucsc.edu/~jor/diol/4524910736.html`.

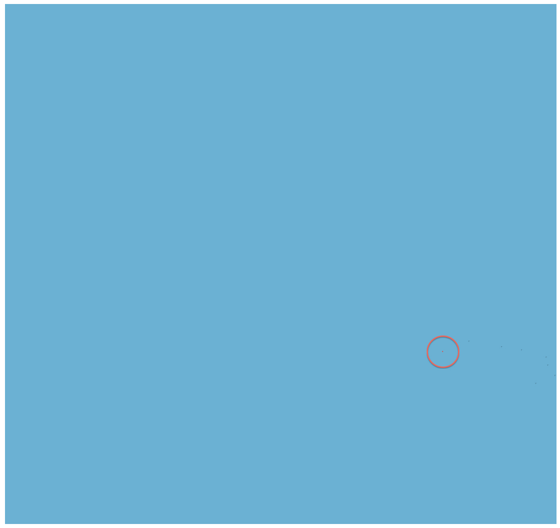


Figure 7.2: A *World* storyworld with only one island. In such a place, nautical travel is futile (and fatal). When he played *Islanders*, Aaron Reed decided to dedicate his life to ship exploration, but died at sea on his first expedition. Upon his character's passing, an omniscient view of the world was revealed and he discovered that his character had been born into a place with only one island—his predilection for the sea had been tragic.

Settlements and Houses

Characters build *houses* on islands, and these are modeled as abstract structures that are positioned on the world grid at fine-grained coordinates that are represented in one-tenth units—for example, a house may be positioned on the plot at `[344.1, 1239.2]`. Houses are named after their builders, whom the system keep tracks of in addition to the year it was built. The essential data associated with a house is its residents (and former residents). As a simulation quirk, houses are permanent and can never be destroyed.

A character *settlement*, then, is modeled primarily as a collection of houses and plots on which new houses may be built. When a settlement is founded by a character, she builds a house at her current position, and the settlement is then attributed one hundred open plots in the surrounding area (unless those plots are already claimed by an adjacent settlement). If at any point all of a settlement's plots are used up, then the characters that live there may decide to expand it. Additionally, settlements may annex one another or become abandoned (if all the characters leave); I will explain how this works below.

Settlements are named similarly to islands: a newly established town may be named for one of its founders, or its name may be randomly selected from the corpus of American municipalities. When a town is named after a founder, that character's surname may be used by itself, or it may be used in a toponymic construction—for example, in Diol, the storyworld that is recounted in the encyclopedia *Diol/Diel/Dial* (the subject of the next chapter), there is a town named for Feodora Quinn that is called 'Quinn Creek'. These toponyms suggest an island topography, but none is actually modeled and the topographical references are selected randomly.

Ships

Characters embark on *ships* to travel on the open ocean. As I will discuss below, this requires a character to carry out the longterm project of constructing such a sailing vessel. Like everything else in *World*, ships are modeled abstractly. While the term 'ship' (and the notion of a ship 'captain') connotes a certain kind of vessel, the inspiration of the Polynesian settlement of the Pacific would suggest something like an outrigger or double-hull canoe [338]. In any event, the physical structure is not modeled.

When a ship is built, it is christened with a name that is randomly selected from a corpus that I constructed by taking all of the words of nine characters or more that were included in a lexicon of the English language; there are over 134,000 entries in this corpus. This was a very simple method, but I found that it produces decent ship names—for example, here is an unfiltered sampling of the first thirteen ships that are listed as being at port in Diol, the storyworld recounted in *Diol/Diel/Dial*:

> Overmourn, Offendedness, Preconduction, Ecclesiology, Xyloglyphy, Karyotype, Unjournalized, Dissoluble, Insulance, Interfilamentar, Emanatist, Emissivity, Cyclocoelous[17]

The system records the builder of a ship, and tracks its position in the world (integer coordinates while at sea or house-style one-tenth coordinates while at port). If it is at port, it will be docked at a position corresponding to one of the coastal coordinates of its port settlement. Additionally, while at sea, a ship will have a captain, that being a character who is responsible for making decisions about the voyage. Below, I will explain how ocean travel is modeled. Finally, beyond the fundamental states of being either at port or at sea, a ship may be in two additional states: sunken or empty (i.e., a ghost ship coasting around on the wind). A ship can only be destroyed by sinking in a shipwreck—as such, ships at port last forever, which enables the evocative situation of an empty island with an immortal ship permanently at port.

Artifacts

There are two kinds of artifacts that are modeled in the simulation: *gravestones* and *ship manifests*. Characters themselves traverse the grid of one-tenth units, and upon dying a gravestone will be placed at the character's final position if there is at least one nearby adult who may place the stone there. A gravestone is modeled as a string containing the character's name, birth year, and death year; if the character who places the stone does not know the deceased character, then

[17] The full listing is available at `https://users.soe.ucsc.edu/~jor/diol/diol.html`, under the heading 'Ships'.

the first two fields will be represented by the placeholder 'Unknown'. Here is an example gravestone:

```
Sybyl Gilles
302-369
```

Additionally, ship captains characters create ship manifests that record information about the initial parties that embark on voyages. These are likewise modeled as strings. As I will explain below, artifacts weather over time and wear away, which is modeled by the simple mechanism of gradually replacing the characters in their string representations with whitespace.

My primary aim in modeling these artifacts was to experiment with *generative environmental storytelling*, a prospect that I discussed earlier and will return to again. In *Islanders*, the player could inspect gravestones (in nearby graveyards) and read old ship manifests (on ships she entered). In one play session that I observed, a friend decided to track down the gravestone of his character's grandfather, to pay homage to him, which turned into a lifelong quest that was driven largely by the information contained in ship manifests (the grandfather had traveled to several islands). Ultimately, the quest was futile, since the grandfather had died at sea (and thus had no gravestone).

While this initial experimentation was rudimentary, I am still fascinated by the idea of generated artifacts that (lossily) inscribe the history of a storyworld. Tarn Adams is now exploring this prospect in terms of *Dwarf Fortress* artifacts [16, 269], and in Chapter 11 I discuss how this idea is being revitalized in *Hennepin*'s artifacts. In a recent paper, Daniel Livingstone and collaborators have written about the phenomenon of expressing the history of a world through (hand-authored) artifacts, a method they refer to as *archaeological storytelling* [694]; this idea is related to the larger area of *archaeogaming*, which is about "archaeology in and of videogames" [994, 345, 993]. More broadly, the game designer Tanya X. Short has called for simulationist game designs that encourage players to investigate simulated material [1150, p. 114-115].

Languages

World's richest modeling appears in its simulation of character *language phenomena*, which is the subject of Section 7.3.16. For now I will mention that language is modeled using an abstract vectorial scheme that ultimately grounds out in the distinct *idiolects* that are maintained by each character.

Whales

As a rather humorous example of a *secret system* [1150, pp. 112–113], *World* quietly features a simulation of *whale* phenomena. Specifically, the system models blue whales who live out tiny abstract lives in a storyworld's ocean.[18] Modeled whale attributes include *age*, *sex*, *weight*, *sexual maturity* (whether mating should occur), *location* (current coordinates), and various attributes pertaining to a whale's *family tree*.

While I intended for the simulation of whales to interact in interesting emergent ways with the simulation of character phenomena—for instance, whaling, the overturning of ships, or situations pertaining to beached whales—I never got as far as integrating the subsystems. As such, the whales go about their little lives in a way that is totally hidden from the rest of the simulation (and human observers). With regard to the tracking of whale family trees, I was specifically interested in the prospect of character and whale genealogies intersecting in interesting ways. For example, I could imagine a variation on *Moby Dick*'s [829] classic tale, where

[18]A note from the code comments: "This class is currently based off of the blue whale. I did not realize this whale could not easily be hunted, so perhaps the gray whale or humpback, etc., should be supported soon".

an Ahab-like *World* character would produce descendants who track the same whale for simulated generations.

7.3 Simulation

Now that I have explained the basics of *World*'s entity modeling, in this section I will outline its simulation procedures. While this simulation engine is more rudimentary than *Talk of the Town* and *Hennepin*, the essence of how I do world simulation has been in place since my early system *Lineage*: create an empty world, have some initial characters move into it, and then proceed from there, timestep by timestep, simulating storyworld phenomena by executing a recurring simulation loop. By this configuration, there are two distinct modes of simulation, one pertaining to the initial setup of a storyworld and the other to its evolution over time—together, these constitute the simulation engine's procedure for *world generation* [751]. I will describe both of these simulation modes in turn.

7.3.1 Setting Up a World

Each *World* storyworld begins with the same essential scenario: there is a large sea scattered with island archipelagos, but no one is there yet, except for a single ship holding several dozen characters, which is guided to land at a randomly chosen island on which they will establish a first settlement. To model this origin story, the system has to do some basic *retroactive simulation*—this entails the creation of an island outside the world, a port settlement on that island, a ship docked at that port, and characters who live in the port settlement. As to the latter, the system probabilistically determines how many characters live at the port, along with the ages of the characters. Some will be generated to also live with additional family members. Once this primordial ship leaves its settlement, it voyages at sea for three years, at which point it is automatically guided to land in the actual modeled world. It is possible for the ship to wreck in those three years, in which case the modeled world will never become inhabited.

Chicken and Egg

This procedure signals a fundamental *chicken-and-egg problem* in this kind of world simulation: unless life itself will be evolved from nothing into something, the simulation must start with characters who were already born at the beginning of story time.[19] As I have just noted, this entails some retroactive simulation (*retconning*) by which each *primordial character* is attributed basic character attributes that are required for the system to operate. In each of my three simulation engines, I have taken the approach of modeling an empty world that a first set of characters then enters. As I explain in Chapter 12, my proof-of-concept pilot episode for *Sheldon County* treats this rather basic technical matter as a *cosmology* that is mythologized in its lyrical narration. For a fascinating approach to retconning, see Jason Grinblat and Brian Bucklew's paper on procedurally generating biographies for historical figures in their dwarflike *Caves of Qud* [427].[20]

7.3.2 Simulation Loop

Beginning with the departure of the primordial ship from its external homeland in the year `0`, the simulation proceeds one story year at a time (since this is the timestep unit), simulating the lives of its inhabitants according to a sim-

[19]In the area of *artificial life*, there are projects that take the latter approach [641]. Of course, these systems do not reach a point where full-fledged characters come to exist in the world.

[20]I use the term 'dwarflike' to refer to videogames in the mold of *Dwarf Fortress*. To me, the extensive use of world simulation separates the project (and others in its wake) from the 'roguelike' genre to which it is conventionally attributed.

ple loop. Generally, this entails the probabilistic triggering of events, where the events that may be triggered at any time depend on the subroutine in which the simulation is currently operating. For example, a subroutine handling character births occurs toward the beginning of the loop, after which character deaths may be triggered, and later still the social simulation is activated. While this yields an implicit ordering of events within a year, there is no modeling of a finer grain of time. In fact, the system itself cannot reason about the ordering of events within a simulated year, though the rendering of event sequences in the prose component of generated encyclopedias—such as *Diol/Diel/Dial*, the subject of the next chapter—expresses the actual ordering of event execution.[21]

In the following sections, I will explain the various phenomena that are simulated over the course of a *World* simulation loop. While these phenomena are grouped into more specific subroutines in the simulation loop, for clarity I will describe them in terms of larger conceptual groupings.

7.3.3 Birth and Death

A character's time in the simulated storyworld is bookended by *birth* and *death* events, whose modeling I will explain in this subsection.

Birth

Characters beside the ones who already exist at the beginning of the simulation enter the storyworld through a process of being born, and all characters depart the world through a modeling of death. If a character becomes pregnant one year, she will give birth to a new character the following year. When this occurs, she and her partner, if she has one, name the new character according to the procedure

[21] This inability to reason about temporal event relations, and moreover causal ones, is fundamental to my critique of *World* in Section 8.4.

outlined in Section 7.2.2.

Death

Death is triggered probabilistically, where the probability of dying from a particular cause is determined by a character and various simulation contexts. Infants may die from birth complications, the elderly may die from natural causes, and a number of other dangers may befall any character: freak accident, disease, starvation (when stranded in the wilderness), ship lost at sea (starvation aboard a stranded ship), murder (more on that below), suicide, and 'mysterious circumstances'. A probabilistic procedure drives the incidence of disease in a character and its communication to other characters (based on factors such as proximity, age, and whether the character is on land or at sea). The probability of character committing suicide depends on her disposition (whether depressive) and also her past trauma: characters who are grieving, divorced, who have murdered, attempted murder (including failed mutinies), or survived a murder attempt. If a character is dying of disease or natural causes, her friends and extended family in the nearby area come to visit one last time. Death under 'mysterious circumstances' is triggered as part of the procedure for catching errors associated with bugs that I could not fix—as a beginner coder, it was easier to smite these characters than to fix certain bugs. Finally, if other characters are nearby, a gravestone will be placed near the character's final position.

7.3.4 Social Simulation

World features an abstract simulation of the social and romantic lives of characters. As characters go about the world, they meet other characters. This occurs according to a simple probabilistic procedure, whereby characters who live with one another are sure to meet, and characters who live (or are traveling) near other characters will have a probability of meeting that depends on various social contexts, namely age, gender, and language.

Social Interactions

Once two characters have met, they may engage in an abstract *social interaction* over the course of a given simulated year; again, this is triggered probabilistically according to various social contexts. As the result of a social interaction, a character may come to consider the other character as a *friend* or as an *enemy*—this depends on the presence of a language barrier (which may probabilistically block friendship or enmity), and moreover on the compatibility of their personalities, which works asymmetrically according to the findings of a study by Maarten Selfhout and others [1126]. I later fleshed out this compatibility method in *Talk of the Town*, so I will describe it in more depth in Section 9.2.5. Friendship and enmity are simple binary relationships (though unidirectional) and they are permanent, except in the case of *Islanders* player characters, who may take actions that ruin friendships or resolve enmities. In addition to friendship and enmity, characters are connected according to familial ties that are specified by the closest relation between the characters.[22]

[22]Here, I say 'closest' because the system tracks character relationships up to the distance of thirtieth cousins. Characters who are born several hundred years into a simulation will have dozens of familial relationships to (nearly) everyone else in the world—for example, a pair of characters may be second cousins; third cousins, once removed; sixth cousins, thrice removed; twelfth cousins six times over; and so forth, all at once. This is because a familial relationship is simply a case of a common ancestor—e.g., being second cousins with someone means that you have a great grandparent in common. *World* characters who are born that far into a simulation will likely have several common ancestors with every other character in the world, with whom they will thus have many relationships. Of course, it is possible for there to be no common ancestry in certain cases of (improbably) extreme isolation. In *World*, this occurs when a subset of the primordial characters immediately leave the first settlement in the world to settle on an isolated island that is never reached by the descendants of the characters who stayed in that first settlement. This is very rare, however, and in most cases *founder effects* [259] lead to everyone being quite related (though decreasingly so according to distance and the dynamics of

Romantic Life

According to a simple probabilistic procedure, a character may become especially romantically attracted to another nearby character. Each year, a pool of nearby characters is constructed for each character according to their sexuality and various social contexts, namely age, marital status, and family relations. A next order of business was to introduce personality compatibility into this reasoning, but I did not find a chance to do that before moving on to *Talk of the Town*. (As I have already noted, character personality was one of my last additions to the simulation.) Once a pool of potential romantic attractions has been constructed for a character, one of them is chosen at random to be the character's primary love interest. If a character already has a love interest, a probabilistic procedure determines whether someone new from the pool will be selected to replace the existing love interest; this depends on concerns such as whether a character is married or grieving the loss of deceased partner. Once a character has a love interest, two social affordances are activated: she may propose marriage or sexually proposition the other character. Again, a probabilistic procedure determines whether these actions will be taken, and in turn another probabilistic procedure determines whether the other character will accept the advance.

---

ocean navigation). While this phenomenon may seem unrealistic, it is only because in the real world we do not have access to an omniscient data store that perfectly tracks the ancestry of all humans—if we did, we would find that you and I, for instance, are related in myriad ways. This is now changing, however, as direct-to-consumer genetic testing [497] enables individuals to easily discover hundreds of fairly close relations. (My dad, who was adopted, has even found two biological siblings through these services.) One function that I developed for *World* computes the *most recent common ancestor* (MRCA) of every character in the world. In the real world, researchers have estimated that humanity's MRCA (the most recent human from whom every living human descends) may have lived as recently as 2,000 years ago [1016]. I learned a lot about genealogy, and phylogenesis more broadly, through its modeling in *World*.

7.3.5 Marriage and Divorce

If two characters decide to marry, then one may take the other's name, and they will begin to reside together (in one of their houses, or on the same ship, or in the same traveling party). When two characters have sex (which co-located married characters do at a high probability), pregnancy (if applicable) may be triggered according to another probability.

Divorce

If a married character has an affair, a probability determines whether her spouse finds out, in which case the latter may consider divorce. Each year, married characters may decide to ask for a divorce, with a much higher probability in the case of a character who was already considering it. If one character asks for a divorce, another probabilistic procedure determines whether the spouse will plead to stay together, and eventually whether they will in fact divorce. Upon divorcing, the characters separate and stop living in the same house (or wilderness campsite), though characters aboard a ship will have no choice but to remain in close proximity on the vessel.

The modeling of divorce in a vaguely historical setting will likely seem curious. I added it because it was easy to model, given what was already modeled, and it yielded genealogical intrigue. Generally, I think it damages the coherence of the simulation domain, but it is in there nonetheless—in *World*'s tiny abstract worlds, characters may divorce.

7.3.6 Foot Travel

Characters who are not aboard a ship may travel across an island by foot. For a character who lives in a settlement, this becomes a matter of deciding whether to leave that place. Foot travel is triggered by a probabilistic procedure that operates over concerns including a character's age, disposition, marital status, family life, romantic prospects in her settlement, whether she grew up in or founded that place, and how long she has lived there. Again, due to its late introduction, character personality does not figure into this procedure, even though a trait like 'openness to experience' would serve as a fitting causal factor.

If a married character decides to travel on foot, she then probabilistically determines whether or not to ask her spouse, which may potentially cause her to abscond in the night. If she asks her spouse, another procedure determines whether the spouse will come along; if the spouse decides not to, then the original character must decide whether or not to continue with her plan, which may lead to a decision to divorce her spouse and take off. Once a character has decided to leave her settlement, she may probabilistically select a nearby town to move to (depending on populations), or she may travel in the wilderness on foot.

Foot travel in the wilderness is essentially a random affair: to determine where the traveler got to after her abstract journey, the system probabilistically determines a direction of heading and a speed of travel. Characters tend to keep their direction, and speed is defined according to a pseudorandom number generated from a normal distribution. Additionally, another pseudorandom number drives movement orthogonal to the character's heading, allowing for northeasterly travel, for example. If a character's computed new position is in the ocean, she is placed at the nearest position to that along the coast of the island on which she is venturing. From here, the character has three options: set up camp in the wilderness, move into a settlement (if any are nearby), or establish a new settlement at her current position (more on this below). As with nearly all character decision making, this is handled by an authored probabilistic procedure.

7.3.7 House Construction

If a character moves into a new settlement, or moves out of her home in a settlement in which she already lives, she will either move into an empty home and build a new one. When selecting a home or plot, a procedure determines whether the character would like to live near the center of town or the outskirts—this should be driven by a character's 'extroversion' trait, but unfortunately it is not—and then she probabilistically picks a plot accordingly. Unlike ships, houses are constructed immediately. Once the character has built her house, it will forever be named for her, though others may eventually come to reside in it over the course of the simulation's entire history.

7.3.8 Settlement Establishment

As I explain below, characters who disembark on a newly discovered island will establish a new port settlement. Additionally, characters who are venturing in the wilderness may decide to establish new settlements at the position of a campsite. This is triggered probabilistically, and when a character decides to start a new town, she will ask any other characters who are camping nearby to join in the effort (who in turn respond probabilistically). Once a town is established, its name is determined according to the procedure outlined in Section 7.2.3.

7.3.9 Settlement Growth, Consolidation, Abandonment

Settlements may grow over time by a simple mechanism. If there are no empty houses or plots in town and a character decides to move into a settlement, then the town will extend into the land around it to establish a new series of open plots on which homes may be constructed. If there is no adjacent land available, due to the presence of nearby settlements, then the characters of the neighboring towns will consider consolidating into a single settlement. Such consolidation is triggered probabilistically, and it may result in more than two settlements merging. When this occurs, the system treats the consolidation as a case of annexation, and a probabilistic procedure determines which settlement will have technically annexed the others (usually the most populous or the oldest). A settlement becomes abandoned if all of its inhabitants pass away or leave town. From this point on, it is considered a ghost town, though it will continue to exist and its houses will remain, which means new characters may come along to reinvigorate it; it may also be annexed by an adjacent settlement.

7.3.10 Shipbuilding

Adult characters in port settlements may decide to build ships. This is triggered by a probabilistic procedure that considers the number of ships that are already at port there, and the population density on the island. If an island is becoming overcrowded (signaled by a scarcity of empty houses or empty plots), characters are more likely to build ships. The process takes multiple years, and the system tracks the progress of each ship as it is being constructed. Characters may decide to abandon these longterm projects (according to a set probability), and a would-be shipbuilder who leaves her project site cannot take the partially completed ship along with her. With each new ship a character builds, she gains in skill, which manifests in a speedier construction rate for the next ship. Once a ship has been completed, it is named according to the procedure outlined in Section 7.2.3 and then docked at the settlement's port. From here, it may be commandeered by anyone, as I will describe momentarily.

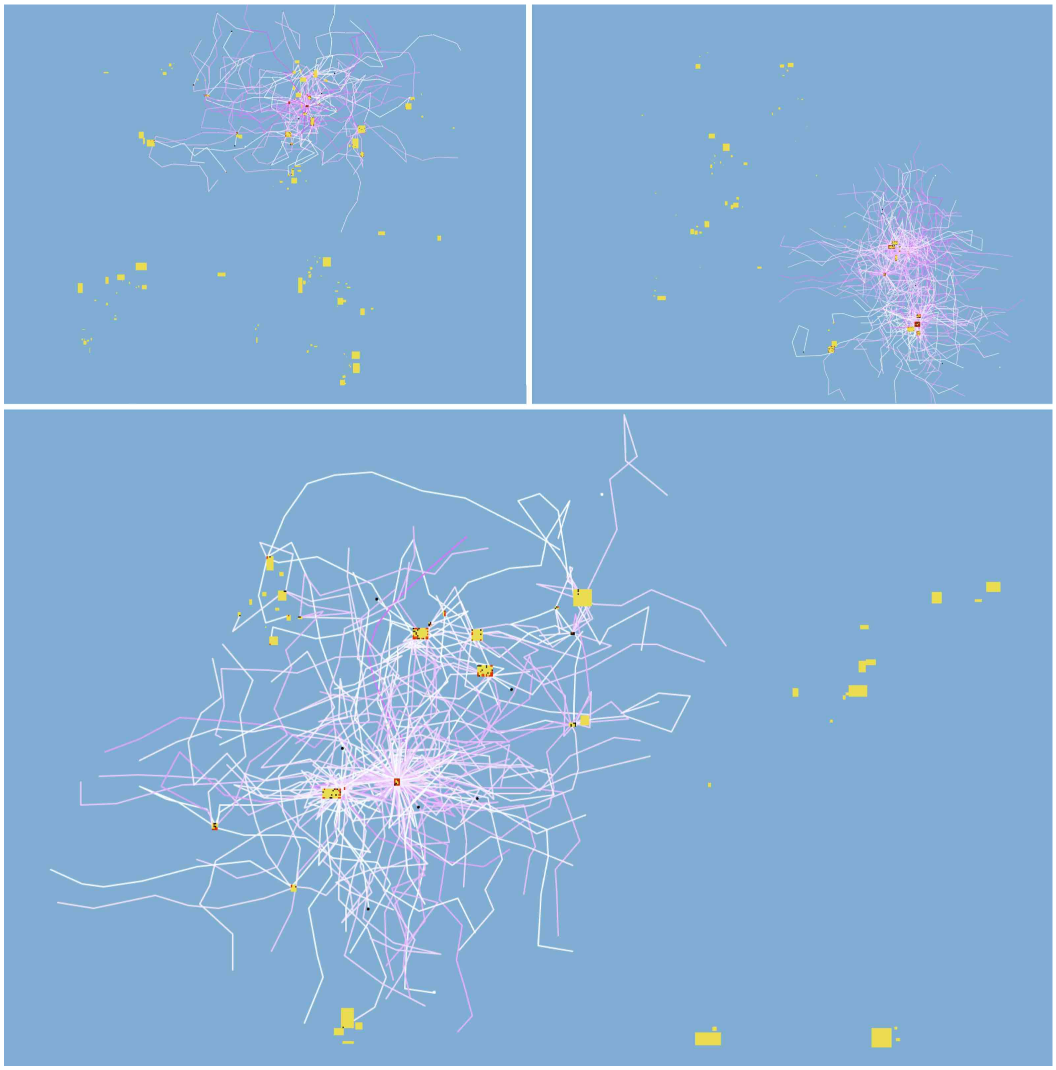


Figure 7.3: A visualization of nautical travel in three *World* storyworlds. Color encodes how recently the route occurred in the world's history (lighter routes are older), and black circles cue the positions of ships that were at sea at the time of visualization. In each case, the routes emanate primarily from the first island settled by the primordial characters. Routes that terminate in the open water indicate that the ship was lost at sea beginning at that point (the routes taken by ghost ships are not shown here); in some cases, ships went hopelessly offscreen, where there is no prospect of encountering land. In the bottom world, an ancient ship improbably reached the archipelago at the bottom of the screen, but the settlement there perished (the black rectangle represents an abandoned settlement) and no ship ever reached it again.

7.3.11 Nautical Travel

When a ship is at port in a settlement, it is available to any character who may decide to commandeer it. Like nearly all emergent events in *World*, ship voyages depend on a probabilistic procedure that considers contextual information including the age of the colony on the island, the time since the ship's last voyage, population densities of the settlement and island, and whether the ship's captain (more on that below) is at port.

Boosters and Captains

If a ship voyage is triggered by this procedure, then a character in the port settlement is randomly selected to be the *booster* for the journey. The booster's first job is to determine who will be the ship's *captain* for the voyage—this is done by selecting the captain for the last voyage, if that character is in town, or else the most experienced ship captain in town, or else the booster herself. Once a captain has been selected, the booster invites everyone in the settlement to join the voyage, and in turn each invitee decides whether to accept by employing a probabilistic procedure that depends on her age, marital status, disposition, and on the prospective captain's nautical experience and history of violence, if any. If two spouses disagree on whether to board a ship, one may ask for divorce, and the one who wants to leave may abscond suddenly.

Before leaving port, the captain authors a ship manifest, naming everyone aboard, which will then exist in the boat as a permanent artifact. Additionally, if the voyage booster has named herself captain but does not have previous nautical experience, the ship passengers may decide to democratically elect a new captain. When this occurs, a procedure of *utility-based action selection* [723, 742] is carried out: every adult on board is scored according to their previous captain experience, and also for their ancestors' captain experience (to support the emergence of intriguing captain lineages). Once everyone aboard has been scored, a probability distribution is fit to the scoring distribution, and a captain is elected accordingly. As I have explained above in Section 4.1.3 with regard to Sheldon Klein's murder-mystery generator, the probabilistic and utility-based approaches to action selection are essentially equivalent from a computational standpoint, though they support conceptually different authoring patterns. I will return to this discussion in Section **??**.

Ship Navigation

Once a ship leaves port, it navigates the open sea randomly, in the style of foot travel, though its captain is viewed by the other characters as being responsible for its navigation. Since *World*'s modeling of time is coarse, this means that the ship will move its position only once a year. To determine a ship's new position, a direction of heading and a speed are probabilistically determined. Ships tend to keep their direction, and speed is defined according to a pseudorandom number generated from a normal distribution. Additionally, another pseudorandom number drives movement orthogonal to the ship's heading, which allows for, e.g., northeasterly travel. If the new position is on top of an island, it will be displaced to be just off of one of the island's coasts. Figure 7.3 visualizes the total nautical exploration of three *World* storyworlds.

If an island is encountered, a probabilistic procedure determines whether the captain decides to stop there; this procedure primarily considers how long the ship has been at sea. After around five years at sea, people aboard may begin to starve. Upon disembarking on a newly discovered island, the characters aboard the ship name the island, following the procedure outlined in Section 7.2.3, and found a new port settlement, where the characters aboard construct houses. If the island has already been discovered, then the ship will probabilistically select a port to dock at according to the populations of its port settlements. If the island has been discovered but is no longer inhabited, the characters will rename it, along with the port settlement at which they disembark. At this point, the characters live in that new settlement and carry out their lives on land. One evocative scenario is the case of an island whose inhabitants abandon it, only to have their descendants return to that ancestral homeland hundreds of years later to take up life in the original inhabitants' settlements and houses.

7.3.12 Crime and Punishment

The prospect of *murder* is driven by probabilistic procedure that depends primarily on the prospective murderer's disposition (whether she is violent), age, and whether she has experienced trauma. If this procedure triggers a murder attempt, then a victim is selected probabilistically from a pool of nearby people; this depends primarily on the potential killer's relationships with nearby people (enemies who are not family members are most likely to be selected). Once a victim has been selected, the murder is attempted, and its result is determined probabilistically—the victim may be killed, or may kill the attempted murderer in self-defense, or the attempt may simply fail.

Blood Feuds

When a character is murdered, someone in her family may become obligated to avenge the killing—as such, there is a cultural of *blood feud* [344] in *World*'s tiny abstract societies. Typically, this onus falls on the closest male relative to the victim, though others may become obligated. When a character has become obligated, she will be likely to attempt to murder the original killer if they are in the same place, but such an attempt is triggered probabilistically according to other concerns (especially her disposition).[23] This simple mechanic of revenge obligation tends to generate full-fledged family feuds: when the killing of one character is avenged, a family member of the original murder will then be obligated to seek vengeance on the avenger, and so forth. In *World*, I have encountered emergent family feuds that lasted centuries and spanned across multiple islands.

Ship Mutinies

Finally, characters at sea may conspire to to carry out *ship mutinies*, which are triggered according to a probabilistic procedure. This may occur when a new captain is elected and one of the candidates feels slighted, especially when the latter has more experience or was the booster who had organized the voyage in the first place. Typically, however, mutinies transpire when a ship has been at sea for multiple years, and especially once characters begin to die of starvation. If someone's loved one starves at sea, she may blame the captain and decide to mutiny. The likeliest scenario for mutiny, however, is the case of a captain deciding to pass up land to continue voyaging, when the safer option would have been to disembark on the sure thing.

Once a character has decided to mutiny, she will attempt to recruit a group of conspirators—whether or not another character will conspire depends on her disposition, relation to the mutineer, and relation to the captain. With the plan in place, an actual mutiny attempt is enabled, but it must be probabilistically triggered to occur. When the attempt is triggered, another procedure determines

[23]A character cannot actually become obligated unless she discovers, through knowledge formation or propagation, that the victim has died. If in turn the prospective avenger discovers that the target of her vengeance has also died, the onus is lifted. I will explain the modeling of such knowledge phenomena shortly, in Section 7.3.14.

its outcome. If the mutiny is successful, the leader either installers herself as a new captain or else decides that the ship will elect someone democratically. If it fails, the captain determines a punishment for the mutineers: if there is a nearby uninhabited island, the captain may navigate to *maroon* them there; otherwise, they will be hung aboard the ship. Upon being marooned, the characters will begin life on the new island, but they will give it a morbid name, like 'Death Isle'. Additionally, marooned characters are traumatized by the experience and become more prone to destructive behaviors. In Section 7.4, I recount an example of marooning in a *World* storyworld.

7.3.13 Artifact Phenomena

As I have noted above, there are two kinds of artifacts that may appear in *World* storyworlds: when a character dies, a gravestone will be laid at her final position (if another character is nearby), and a ship captain creates a ship manifest (listing the names of all the passengers) upon embarking on a new ocean voyage. Each of these artifacts may wear away due to the passing of time. Artifacts are represented simply as strings, and weathering is modeled by procedurally replacing characters in those strings with whitespace. As such, a gravestone might weather in this way over several centuries:

```
Sybyl Gilles
302-369

S b l Gil  s
3 2-36

S   l Gi   s
3  -
```

Again, this weathering was meant to provide intrigue for *Islanders* players, who could inspect nearby artifacts as they moved about the world.

7.3.14 Character Knowledge

Toward the end of my development work on this project, I began to experiment with the modeling of character knowledge phenomena, particularly *knowledge propagation* [197]. As I explain in Chapter 9, this initial exploration would become the basis for *Talk of the Town*, my second simulation engine, which is primarily about character knowledge phenomena. Since the method employed in *World* represents an embryonic precursor to the approach developed for that engine, I will hold off on discussing related work until the later chapter.

Knowledge Formation

In *World*, characters keep track of the last known whereabouts of other characters with whom they are acquainted. This information is modeled as a simple tuple of the form (who, when, where, status (living or dead), source, source date). As I noted above, characters may interact over the course of a year, depending on factors including their languages, relationship, and so forth. When characters interact, they keep track of where and when this happened, and store this information as whereabout tuples. For example, if in the year 321 a character named Llewellyn Vaclav interacted with another character named Gavriel Guillermo in a town called Minneapolis, Llewellyn would update her knowledge to include a new whereabouts tuple structured like this (angle brackets denote pointers to the actual objects representing those entities, which resolves referential ambiguity): (<Gavriel Guillermo>, <Minneapolis>, 321, 'living', <Llewellyn Vaclav>, 321). Further, when someone dies, all the other characters in that location will remember that the person died there and then.

Knowledge Propagation

I have just described the ways in which character knowledge may be *formed*, but it can get passed on to other people as well. During an interaction, characters bring up other characters with whom their interlocutors are also acquainted. If one of the speakers knows more recent whereabouts for this person, the other will update his or her own knowledge to reflect this information. Thus, for example, if during that interaction Llewellyn Vaclav told Gavriel Guillermo that a common acquaintance named Scarlett Vaclav had died aboard the ship Preconduction in 298, Gavriel would update her knowledge to include the following entry: (<Scarlett Vaclav>, <Preconduction>, 298, 'dead', <Llewellyn Vaclav>, 321).

Though I did not implement this, I also intended for character knowledge to update upon the inspection of artifacts. For example, if a character encountered a gravestone in the wilderness with her friend's name on it, she would know that her friend had died there in the year listed on the gravestone. Likewise, ship manifests could impart the whereabouts of passengers on the ship during the year of departure. Of course, there is possible ambiguity, in that multiple characters may have the same name, and weathered artifacts may lead to uncertainty about the name on a gravestone or manifest. Because I had no means for representing memory fallibility, a core concern in *Talk of the Town* [1061], I held off on taking on this challenge, and ultimately I never got to it in *World*. In *Hennepin*, the subject of Chapter 11, I am now revisiting artifact simulation, with the modeling of artifacts that trigger knowledge updates upon being inspected. I explain how this works in Section 11.2.11.

Forgetting

Finally, knowledge may also be forgotten due to the passing of time. This happens probabilistically, according to the amount of time that has passed and also a character's memory attribute. As I mentioned above, a character's capacity for memory decreases over time, and as a result she may forget things she once knew. Because the only kind of character knowledge that is explicitly modeled is the last known whereabouts of other characters, forgetting pertains exclusively to such information. Specifically, a character may forget about the whereabouts of another character, in which case the corresponding entry is removed from her knowledge. This produces the uncanny (yet evocative) image of an elderly character who gradually forgets about everyone she knows, one by one.

7.3.15 Character Internal Worlds

One characteristic of my simulation practice is an emphasis on simulating the internal lives of characters, a predilection emerged at the end of my development work on *World*. I am now engaging this phenomenon more deeply in *Hennepin*, and in Chapter 11 I situate my current approach against related work.[24]

[24] In a recent email exchange, Chris Martens offered her thoughts as to the centrality of character interior life in narrative (and the peculiar disregard for it in computational narrative): "While I think it's well-understood in the narrative generation community that prevailing techniques address "plot" much more than (or to the exclusion of) "discourse," I think that even within the realm of plot, computational narrative is too fixated on observable actions as the central *driver* of plot. I don't believe this fixation reflects the reality of what makes popular stories *function* as stories. [...] One concrete hypothesis I have is that if one were to comb through any popular novel, you would find that the vast majority of text doesn't describe "actions" at all but rather descriptions of characters' changing interior attitudes and responses to events, and that descriptions of actions themselves often serve only to add depth and detail to a character, not to advance the plot" (personal communication, April 7, 2018). I love this argument and agree that work in computational narrative should better emphasize character internal worlds.

Grief and Regret

As I have mentioned above, characters may grieve the loss of loved ones who they know to be dead—when grief is triggered, it puts the character into a grief state that endures probabilistically according to a yearly probability of the grief subsiding. Characters in a grief state are traumatized, which affects decision making in the ways I have outlined above. When a character travels to somewhere where there are no other characters or potential partners, or where no one speaks her language, she may lament her decision to leave an earlier place of residence.

Wonder, Worry, and Reminiscence

Another internally oriented action concerns characters who wonder and worry about loved ones who live elsewhere; this is triggered probabilistically according to the character's neuroticism trait value. To actually take this action, a character searches through her knowledge to select a loved one whose last known whereabouts (from the worrying character's perspective) corresponded to the departure of a ship (which may have been observed by the worrier) or residence in another place (which depends on knowledge of such residence propagating back to the worrier). In *World*'s generated storyworld encyclopedias, such actions are recounted in the following way: "Gianna Shumeet is wondering about her husband Capt. Addie Shumeet, who 2 years ago left for a voyage on the Microbarograph and hasn't been heard from since".[25]

Finally, characters may reminisce about past events. When this action is triggered, a character abstractly recalls her time in a former place of residence, from at least five years into the past, for which she was at least five years old at

[25] This sentence was taken from the *Diol/Diel/Dial* entry for a character named Gianna Shumeet, which is hosted online at `https://users.soe.ucsc.edu/~jor/diol/5544165584.html`. As his own entry explains (see `https://users.soe.ucsc.edu/~jor/diol/5505259280.html`), Capt. Addie Shumeet had in fact just been killed in a mutiny aboard the ship.

the time of residence; here, ship voyages are preferred, since I find them to be inherently more evocative. While grieving puts a character into a new state that affects her future behavior, the other internal phenomena are generated simply for narrative intrigue (since they may be expressed in a storyworld encyclopedia).

7.3.16 Character Language

The most ambitious subsystem in *World* pertains to its modeling of character linguistic phenomena, which I will outline in this section.[26] At the start, all characters in the storyworld are together on the primordial ship and speak the same language, but over time this language will change and possible diverge into new languages, as I will now explain.

Overview

Each character has an *idiolect* (for each language they speak) that is represented by a sparse binary vector composed of 1,000 arbitrary bits. To account for linguistic innovation, bits in this vector may be flipped at a certain probability (which changes over the course of a character's life and is highest during adolescence). When characters speak to nearby people over the course of a year, they may adapt their idiolect to be (slightly) more like their interlocutors' idiolects—this is modeled by flipping a bit or two in the vectors.

These simple mechanics are enough to cause dialects, and eventually new languages, to emerge: characters can only talk with others who are nearby (in the same settlement, in most cases) and most people do not move across the world, which means that arbitrary linguistic innovations (bits that were flipped) occurring in some place are only likely to be adapted by the inhabitants of that place.

[26]This work was first reported in a workshop paper on the subsystem [1037]. The prose here is lightly revised from that source.

Young children then inherit (initially) an idiolect that resembles their parents' idiolects, which reinforces the linguistic quirks of a particular area.

In this subsystem, dialects are represented compositionally by taking, for all 1,000 bits, the majority bit value given all idiolect vectors among the speakers living in some settlement. Likewise, a language is represented in this same way, but by instead taking into account the idiolects of all its speakers everywhere. If in some year any two dialects of the language become so dissimilar (measured by the number of bit values that do not agree) as to exceed some threshold, the dialects are reconstituted as full-fledged (sister) languages.

Related Work

Before discussing this subsystem, I would like to outline some related work, particularly in the area of computational media. Many videogames feature fictional *storyworld languages* that are spoken by characters in the games' diegeses [961, 952]. These languages are typically represented using constructed sound systems or orthographies, often runic-looking ones, that are intentionally opaque to the player. This design move has been employed for light worldbuilding that is nonessential to gameplay, like the variant of Hylian used in *The Legend of Zelda: The Wind Waker* [881], as well as in support of abstract procedural dialogue (thereby skirting the cost of full natural language generation), a purpose served by the Simlish language of the *Sims* series [952]. Other games, however, use storyworld languages in their core gameplay. In the *Myst* series, for example, the fictional D'ni language is central to certain puzzles [952]. Chris Crawford's innovative *Trust & Betrayel: The Legacy of Siboot* [221] utilizes an early version of the designer's modular iconic language, Deikto [223], as an important player interface. Exxos's *Captain Blood* [328] employs a similar modular language that likewise allows players to construct novel messages that are understood by the system [961, p. 53]. *World of Warcraft* hinders communication between players whose characters are not in the same factions, using the narrative conceit that the factions are associated with mutually unintelligible languages [706].

But while generative methods have been employed to produce a huge variety of videogame content [1134], including content representing other sociocultural concerns [548, 523, 425, 784, 203], I am not aware of any project that has procedurally generated storyworld languages. Even *Dwarf Fortress* [17], whose range of content generation is famously vast, does not generate or alter its languages.

Outside of videogames, but still within the purview of computational media, there have been some interesting experiments in this area. As he explains in a detailed write-up [899], Martin O'Leary's *Uncharted Atlas* [900] is a *Twitter bot* [638] that utilizes a subsystem for constructing *naming languages* [476] that are used to name locales in the bot's procedurally generated world maps. In a similar vein, Kate Compton's *Kambamanx* [202] uses *Tracery* [204, 205] to generate poems in generated languages.[27] Finally, though not a computational project, Thorny Games's *Dialect* [1250] is a board game about language change.

More broadly, I should situate *World*'s language subsystem against the application area of artificial intelligence that computationally models language phenomena using agent-based simulations [1193, 406]. In this body of work, there is a particular focus on operationalizing theories about the origins of language [1192, 1193] and about specific attested linguistic phenomena—for example, the emergence of *vowel harmony* [456] or *grammatical agreement* [106]—where the simulation outcomes serve as empirical results for the operational theory. This mode of scholarship is what Joshua Epstein has termed *generative social science* [314].

[27]Polymath Vi Hart has rendered some of Kambamanx's outputs in song and posted audio recordings online. To hear them, follow the links included in this Twitter thread: `https://twitter.com/GalaxyKate/status/858396760124919808`.

Relative to this craft, my method here is quite simplified in its modeled evolutionary processes and linguistic representations, but it does appear that this project could be novel in its simulation of diachronic (across centuries) macroscale (the phylogenetic level) language change in a massive discrete physical space (whereas other work typically models a single abstract agent environment).

Finally, let us welcome back to the limelight Sheldon Klein, a main character from the first half of this dissertation. A few years after I implemented *World*'s approach to simulating character language phenomena, I learned that Klein had developed a remarkably similar system fifty years prior [585, 588, 589, 599, 591]. Moreover, this particular simulation framework ultimately developed into his murder-mystery generator, discussed at length in Chapter 4, as he explains in a brief retrospective [46]. In Klein's unnamed language simulation, developed approximately between 1964 and 1974, agents in abstract speech communities converse with one another over the course of a few simulated decades. Each speaker maintains a both a generation grammar and a recognition grammar, and conversation works something like this:

> During the course of a conversation, one individual will generate a form and another will attempt to parse it. Should the parser's rules be inadequate for the task, he may borrow the necessary rules from the generation grammar of the speaker, and perhaps use it when it is his turn to speak. [588, p. 2]

Klein's agents are modeled according to a number of concerns—"age, sex, village, clan, religion, household, marital state, work groups, and social status" [599, p. 8]—and a loose social simulation determines who will interact with whom, in a way that depends on these attributes. As the simulation proceeds, agents die and new ones are born from existing ones, leading to *language acquisition*. Furthermore, agents can move between speech communities, which generates second-language phenomena and *language contact*, a topic to which Klein dedicated a paper on the project [591]. In another article, titled "Monte Carlo Simulation of Language Change in Tikopia & Maori" [599], he reports on an experiment in operationalizing a theory pertaining to specific real-world languages.

While a more extensive excavation is in order, it is worth noting already that Klein's language simulation may be the earliest foray into a number computational areas that are central to the concerns of this dissertation: language modeling, of course, but more importantly, world generation and social simulation. This system is really important. In terms of its relations to *World*, both frameworks model a number of linguistic phenomena—namely native acquisition, non-native acquisition, and change—and do so by taking a fine-grained agent-based approach. Critically, however, the systems differ in the in the way that language itself is represented: while I utilize an abstract vectorial scheme, Klein remarkably implements full-fledged grammars for both generating and recognizing, which means his simulated conversations are in surface-level natural language.

Background: William Labov's Incrementation Model

Broadly, and at times specifically, my approach to simulating character language phenomena in *World* is an operationalization of William Labov's *incrementation model*, which describes real-world language change as a bottom-up evolutionary process that is driven by the speech activity of individuals [633].[28] While I will periodically explain more details below, the thrust of the model is that the individuals composing a new generation of speakers begin life by inheriting the speech tendencies of their parents, which they proceed to adapt and modify during adolescence, before eventually transmitting their specific innovations to their children, the next generation of speakers, who carry out this process again; Figure 7.4 illustrates how the model works in *World*.

[28]William Labov is the sociolinguist whose work on *tellability* I outlined in Section 4.1.1.

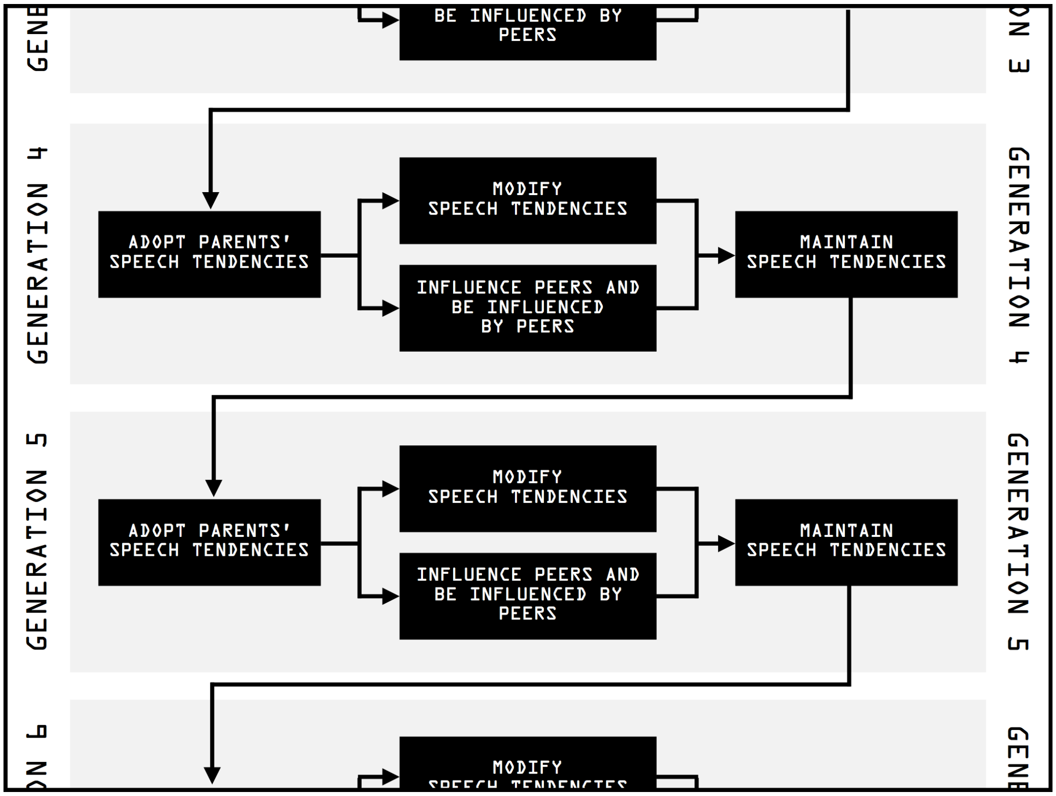


Figure 7.4: An illustration of William Labov's *incrementation model* of language change as it is operationalized in *World*. The characters of a new generation in a speech community begin life by inheriting the speech tendencies of their parents, which they proceed to adapt and modify during adolescence, before eventually transmitting their specific innovations to their children, the next generation of speakers, who carry out this process again. Over time, this cycle evolves a language and drives its dialects apart: because different mutations will obtain in different speech communities, the speech tendencies of those communities (associated with dialects) gradually diverge over time. Eventually, such divergence may become so substantial that speakers of related dialects become *mutually unintelligible* to one another, at which point it is more useful to say that one or both have evolved into *daughter languages* of the language for which they were previously called dialects. In *World*, a storyworld's language family tree grows as agent-level speech tendencies mutate over the course of simulated centuries.

Labov's work extends two earlier notions in linguistics, which I would like to identify briefly. Leonard Bloomfield identified the *speech community* as the fundamental human group:

> A speech-community is a group of people who interact by means of speech. All the so-called higher activities of man—our specifically human activities—spring from the close adjustment among individuals which we call society, and this adjustment, in turn, is based upon language; the speech-community, therefore, is the most important kind of social group. [119, p. 42]

Critically, a speech community may be subdivided into smaller communities whose members engage more frequently in speech interaction. In this way, a speech community comprising all the speakers of a language may be subdivided into smaller communities corresponding to dialects and smaller groups, and so forth, perhaps even to the level of considering an individual as a kind of speech community [443]. Bloomfield's notion forms the bedrock of sociolinguistics, and it suggests a bottom-up mechanism of language change rooted in the speech activity of individuals, which is the thrust of Edward Sapir's notion of *drift* [1092]: language change is a gradual process built up by small changes over time.

Labov unifies these ideas in a coherent model that identifies the nature of incremental language change as being grounded in the activity of individuals in a speech community. Generally, this kind of model is referred to as characterizing *change from below*, as opposed to change under pressure from external (or 'above') forces, which occurs in *language contact* [1249], for example. The phrase 'change from below' first appeared in a 1965 paper by Labov [628].

Vectorial Representation

My approach utilizes a vectorial scheme by which a language is represented as a sparse *bit array* with 1,000 entries—i.e., an array composed primarily of zeros but with a small number of ones. In this representation, bits stand for abstract linguistic features. This level of abstraction is targeted because it supports the efficient simulation of bottom-up language change among many characters, as well as the computation of mutual intelligibility between any two characters; as I explain below, these are the two core concerns of this subsystem. While languages are represented in this way during the simulation of language evolution, I was interested in the prospect of using underlying language vectors to generate concrete *surface-level* representations of those languages (i.e., *conlangs* [184, 895]). This was never implemented, however.[29]

Idiolects, Dialects, and Languages

As in the real world, characters in *World* storyworlds instantiate their own *idiolects*, and the idiolects of multiple agents work in tandem to compose *dialects* and full-fledged *languages*. An idiolect characterizes the peculiar linguistic patterns of an individual, and a dialect represents the linguistic patterns of a speech community. In *World*, dialects are associated with groups of characters who speak the same language and live in the same settlement or on the same island (i.e., there are settlement dialects and also island dialects); both are represented using the same vectorial scheme discussed above. All three of these linguistic systems are examples of *varieties*, a useful term from sociolinguistics that I will utilize below.

As I have noted, language change in *World* originates in the propagation of arbitrary linguistic innovations at the level of idiolects (modeled as bit flipping). Change at the level of dialects and languages, then, is merely a byproduct of id-

[29] In my workshop paper on this subsystem, I wrote this about the prospect: "Another area of future work [...] is the prospect of generating fully specified *conlangs*, or constructed languages, from them. We imagine that this would involve treating a vector as a genetic sequence that, by some mapping scheme, yields phenotypic traits that express in the surface representation of the language or in its rules. By this approach, related languages would produce similar conlangs, since they would have similar underlying vectors and would be generated using the same mapping scheme. For this project, language vectors would likely need to be extended past the 1000 entries that we used in this paper. This is because important language features—e.g., the use of gendered pronouns—would likely require many bits, if only for the practical reason that the flipping of a single bit in a language should only correspond to a small surface-level change" [1037, p. 7].

```
00001000000000100000000000000000000000010000001010
00000100001000000000000000000000000000001000000000
00000000000101001000001000000000000100000001000000
00010000000000100000010010001000000000001000001010
00001000100000000000000000000010001000000000000000
00000000001010000000000000000000000000001000000000
00000000000010000000000000000000001010000000100000
00000000001000000000000001000000000000000000000000
00000000000000000000000000000010001001010000000100
10000000000000000000000000000000000000000000000000
00000000000110000001100000000000000000000000000000
00000100000000110011000000100100000000000100001000
00000000000100000000100000001000000000000110100000
00000001000010000001001000100000001001000000000000
10000000000001000000000000001000101000001010000000
00110000100001000001000000001000000000000000001000
00100000000000000000000000100000100010000000000000
00000000000000000000010000000000000000000000000000
00000000000000100000000011000000000100010100100000
00000000000000000000001000101000000001000000000000
```

Figure 7.5: A *World* language. In its simulation of character language phenomena, *World* utilizes an abstract vectorial scheme whereby instances of linguistic *varieties*—idiolects, dialects, languages—are each represented as a string of 1000 zeros and ones. Over time, a speaker adapts her speech tendencies by probabilistically flipping bits in her idiolect vector (pending social concerns), and such mutations may be propagated to others in her speech community (pending social concerns). A vector representing the form of a dialect at some point is constructed by taking the most frequent bit value for each entry, given all the idiolects of the speakers in the associated speech community (either a settlement or island). Language vectors are constructed in this same way, except by taking into account the idiolects of all of the speakers of that language across the entire storyworld. If at any point some dialect differs from its associated language by more than 25 bits, it will be reformulated as a distinct *daughter language*. Over time, a storyworld's language family tree grows by this process, but critically this macro-level change is driven bottom-up by concrete micro-level changes enacted by diegetically situated agents. William Labov refers to this phenomenon as *change from below*.

iolectal change, which is tracked at infrequent time intervals. Specifically, once every story year, the system updates the vector representing a language by surveying the idiolects of its proficient speakers to assemble a new vector whose entries contain the *majority bits*, across all speaker idiolects, for each bit index. For example, if 500 speakers of a language have a `1` as the 100th entry of their idiolect vectors and 499 have a `0`, the language would take a `1` for that entry of its own vector (since it is the majority bit across the language's speakers). Dialects for each language are similarly instantiated for each settlement and island where the language is spoken, though of course these procedures only operate over the idiolects of speakers living in the relevant areas.

Language Acquisition

Children in *World* acquire language from parents and other individuals in their homes, and additionally from peers once they begin leaving their homes to interact with other characters living in their settlements. Additionally, adults may acquire non-native languages given sufficient exposure to them. In both cases, *fluency* is represented as a floating-point value between `0.0` and `1.0`, where `0.6` is treated as a threshold for *proficiency*.

Up until puberty, characters in *World* may acquire native-level fluency in any language to which they are sufficiently exposed; this operationalizes the *critical period hypothesis* in linguistics [676]. The *acquisition rate* (rate at which fluency is acquired) for characters gradually decreases until age fourteen, and thereafter steeply. Prior to the age of four, characters will only be exposed to languages spoken in their homes, specifically languages that are spoken proficiently by residents in the home. If there are multiple such languages, children may or may not acquire multiple or all of them—this is determined probabilistically, where languages spoken by parents and/or by multiple people in the home are more likely to be acquired. Upon reaching the age of four, children begin to leave the home and interact with other characters living in the same settlement. Here they may be exposed to many languages, which they acquire (or gain fluency in) probabilistically, where the chance of picking up a language is commensurate to the percentage of characters in the settlement who are proficient in that language. This probability is also boosted according to the degree that a language is spoken by friends of this character. To initialize a child's idiolect for a newly acquired language, she simply inherits the current idiolect of someone in the home (most likely her mother, per the incrementation model [633]), if it was acquired in the home, or else the vector for the current dialect of the language in her settlement.

After puberty and until middle age, agents may acquire non-native languages at a gradually diminishing acquisition rate. Upon middle age, this acquisition rate begins to decrease exponentially. To initialize an agent's idiolect for a newly acquired non-native language, the system operationalizes the notion of a *foreign accent*. This is done by taking the local dialect of the language being acquired and warping each of its bits (at a 20% chance) to match the bit in the corresponding index of the agent's first language (i.e., the language she is most proficient in).

If an agent does not have access to other speakers of a language that she speaks, she may lose proficiency in that language from disuse. This often occurs when someone leaves a home island for new environs in which no languages already known by the agent are present. Specifically, language fluency decays at a rate of 3% per year of non-exposure.

Language Change

Following Labov's incrementation model, *language change* in *World* works bottom-up according to the accumulation of minute changes occurring at the level of idiolects. This process is characterized by discrete linguistic *innovations* (modeled as bits flipping) that propagate, by a process called *adoption*, across speaker groups as agents incorporate these new forms into their own idiolects. Because dialects and languages are composed according to the idiolects of their speakers, idiolectal mutations will produce changes in these larger varieties as the former propagate widely.

Similarly to biological evolution, dialects and languages change as discrete mutations in the speech patterns of individuals propagate across speaker groups [226]; the processes driving and governing linguistic mutation, in particular, are referred to by the banner term *innovation* [838]. In *World*, innovation is abstractly modeled as *bit flipping*; that is, a discrete linguistic innovation by an individual is simulated as the flipping of a single randomly selected bit in her idiolect vector. Each year, each agent in the storyworld has a probability of flipping a single bit, which is determined by the agent's age and personality. Specifically, agents between four and seventeen years old have a chance of innovating that approaches 60% (an operation of the incrementation model, which posits this age range), and thereafter the chance decreases as age increases and bottoms out around middle age. Additionally, the chance of flipping a bit depends on gender (the incrementation model asserts that girls and women are the primary drivers of language change; [630]) and speaker personality, with more extroverted agents being more likely to innovate; the latter operationalizes accounts of personality influencing innovation [299].

Linguistic innovations propagate within and across speaker groups by a process called *adoption* [630, 633]. In *World*, I model adoption as the unconscious flipping of bits in an agent's idiolect vector to match the corresponding bits in the idiolects of nearby agents. This is done for each language that the agent speaks, and works in the following way. Once a year, the system collects for an agent all of her friends that live in the same settlement (and speak the language at hand) and, for each of these friends, determines a *power of influence*. An agent's linguistic power of influence over another agent is determined by the influencer's proficiency in the language under consideration (more proficient speakers are more influential) and the age difference between the two agents (influence is decreases with age difference). Having this, the system iterates in a lockstep manner over the agents' respective idiolect vectors, and, for each instance of a mismatch between the vectors, flips the bit in the idiolect of the agent who is being influenced at the probability of the derived power of influence. For example, if the power of influence for the pair of agents is 0.5, each mismatch between their idiolects would have a 50% chance of being resolved.

Innovations spread across friends close in age to match the incrementation model's account that adoption primarily occurs within peer groups (who then pass on the innovations to their children), but I should note that this fails to capture the pivotal role of social stratification in this process (since *World* does not model these concerns) [627, 298, 630].

Language Birth and Death

Over extended periods of localized innovation and adoption, dialects tend to grow apart as a function of geographic distance [633]. Eventually, such divergence may become so substantial that speakers of related dialects become *mutually unintelligible* to one another, at which point it becomes more useful to say that one or both have evolved into *daughter languages* of the language for which they were previously called dialects. This is *language birth*, and in *World*, demarcation proceeds as follows. As I noted above, dialects and languages are reconstructed yearly. Once this is done, the vectors for each island dialect are compared against the corresponding language vectors, and if any difference of more than 25 bits is encountered, the pertinent dialect is reified as a new language.

As this process repeats, the storyworld's *language family* becomes more varied, and a phylogenetic tree characterizing it grows accordingly—just as *World* tracks genealogical information about characters, it does so with languages as well. I should also note here that the storyworld's *primordial language* (the language that already exists at the beginning of story time) is generated by turning its bits on at a 10% chance. All speakers alive at that time simply begin with that vector as their respective idiolect vectors. Additionally, the system constructs two *protolanguages* [837]: the mother language of the primordial language (originating on the island outside the world that the primordial characters depart) and that language's mother language (which allows the language family tree to reach all the way back into prehistory).[30]

Finally, *language death*: when the last speaker of a language dies or loses fluency in it, the language vanishes with her.

Missing: Language Contact

A major modeling gap in this subsystem is the lack of any simulation of *language contact* [1249]. This is the phenomenon of disparate linguistic varieties converging on small geographic areas, called *contact zones*, in which people who cannot speak to one another have to communicate for commerce or other reasons. From such contact might emerge a *pidgin* language, which is characterized by an underspecified recombination of features of the languages in contact. When a generation of children are exposed to a pidgin, however, they adopt it as a native language and invariably flesh it out to full specification, at which point it becomes a *creole*. Though I never got to it, I planned to model contact by having the system build *pidgin vectors* that would recombine subsets of the contact language vectors, which could then be filled out to full-length creoles using random bits.

[30] The emergent language family tree for one *World* storyworld may be explored starting from the *Diol/Diel/Dial* entry for Thal, the protolanguage for the protolanguage of the world's primordial language: `https://users.soe.ucsc.edu/~jor/diol/4513623632.html`.

Language Names

A language is named in one of a few different ways, depending on the nature of its origin. The prehistoric protolanguage at the root of a storyworld's language family is named by selecting randomly from a corpus of language names, which I constructed by procedurally recombining a number of short letter sequences. This language's only known daughter language is the one that originates prehistorically on the off-world island from which the set of primordial characters originate, and its sole known daughter language is the actual storyworld's primordial language. The latter is named by selecting an element from the language names corpus, and the former is named by prefixing `Proto-` onto the name attributed to its daughter.

From here, languages are named according to a probabilistic procedure, as they emerge, by which there is a chance of either selecting from the corpus again or generating a demonymic name derived from the island on which it is primarily spoken. In the latter case, a simple function considers whether the island name ends in a vowel—if it does, the language name will end in `-an`, otherwise another suffix (e.g., `-ic`) will be probabilistically selected from a list of options. For instance, a language emerging on an island called Niota would be named Niotan.[31]

World Toponyms

Finally, while my representational scheme is abstract, as I have explained in the preceding sections, there is one surface-level linguistic unit that *is* produced by each language: a name for the storyworld. For the primordial language spoken by its first arrivals, a world toponym is randomly selected from the corpus of language names. (The ancestors of the primordial language do not have a name

[31]Niotan is in fact an obscure language that emerged in Diol, the subject storyworld of *Diol/Diel/Dial*. As its encyclopedia entry recounts (see `https://users.soe.ucsc.edu/~jor/diol/6363583056.html`), its origin island Niota became uninhabited, and accordingly the language (which was only ever spoken by seventeen individuals) became extinct.

for the storyworld because they originate outside of it.) From this point on, a newly instantiated daughter language may mutate the world toponym of its parent language—this is triggered probabilistically and operates according to a simple procedure that models common sound changes that have been identified in historical linguistics [837]. In the subject storyworld of *Diol/Diel/Dial*, for instance, this mechanism led to a gradual mutation of world toponyms from `Diol` to `Diel` to `Dial`, though most languages retained the original variant.[32]

7.3.17 Whale Life

As I noted above, *World* secretly contains a hidden simulation of the tiny abstract lives of whales who reside in storyworld oceans. By this simulation, a series of primordial whales are generated at the beginning of story time and placed near either a *feeding ground* or *breeding ground*, the positions of which are determined procedurally. Each year, sexually mature whales travel between the feeding ground and the breeding ground, while calves stay near the feeding ground. This navigation works similarly to character on-foot travel, except the direction of heading is rigged to point roughly toward the targeted destination.

Modeled whale actions (beyond swimming) include mating, giving birth, nursing a calf, and eating fish (this is modeled abstractly by incrementing the whale's weight). Additionally, whales who swim near islands may become beached on them (according to a set probability), but unfortunately characters at such locations do not react to such whales.

As I explained above, this subsystem was implemented late in *World*'s de-

[32]A few *Diol/Diel/Dial* encyclopedia entries express this evolution: the primordial language Aniumustpyrx refers to the world as 'Diol', while its daughter language Pelicanic mutated this to 'Diel', which its own daughter Guillermoan transformed into 'Dial'. This can be seen by starting from the encyclopedia entry for Guillermoan (and following the ancestral path back to Aniumustpyrx): `https://users.soe.ucsc.edu/~jor/diol/7683358032.html`.

velopment, and I never got all the way to fully integrating it with the other subsystems in the simulation engine. As a result, emergent intrigue surrounding potential interactions between characters and whales—characters watching and hunting whales, or whales overturning ships and beaching at ports—cannot actually obtain in a storyworld. For this reason, whales are not included in the project's generated storyworld encyclopedias.

7.4 Emergent Phenomena

As I discussed at length in Section 4.1.1, a first question to ask with regard to a simulation engine is whether it can even produce stories at all.[33] Though its modeling is rudimentary and often suspect, as I will discuss in the following section, *World* still generates narrative potential in the form of evocative *emergent phenomena*. At times, this emergent material takes the form of a full-fledged event sequence, but more often it is a poetic image that is constituted in a brief moment in time. In this section, I will outline a few examples of emergent stories and images that I documented during my development of the simulation engine.

An Old Friend Appears

In my initial testing of the newly implemented system for handling character knowledge phenomena, I encountered an interesting dramatic nugget. A character who grew up on an island with a small population left it, at age nineteen, on a ship bound for a large, populous island that he would live on for the rest of his life. Decades later—by now the man was in his late 60s—another ship from his home island arrived there. It was the first ship to do so since the one that he had

[33]In that section, I identified a boringness continuum that spans from non-stories to boring stories to good stories.

journeyed on all those years ago. Among the passengers of this vessel, which took up port at the settlement in which he lived, was a childhood friend of the man.

The two caught up immediately and began to discuss the family and friends that the man had not seen in decades. As he would find out, most of these characters had died in that span. This was an uncanny and evocative image: a man in his 60s hears about the time and place of death, one after another, of everyone that mattered to him in his adolescence. Critically, he discovered that his immediately younger brother was murdered in their hometown *the year after he left*. As I explained in Section 7.3.12, in *World*'s storyworlds an onus is placed on a close relative of someone who is murdered to avenge that person's death. Even though the crime had been perpetrated nearly fifty years ago, by finding out about it now the man caused the simulation to trigger this onus mechanism, and so an elderly man became obligated to avenge his brother's killing more than four decades after it occurred. Later in the interaction, however, the man discovered that the murderer had also died a number of years ago, and so the onus was lifted almost as soon as it had been asserted.

While this procedure, outlined in Section 7.3.14, pertains to the simple updating of a character's knowledge as to the last known whereabouts of other characters, my human mind could easily discern the narrative potential in the raw data. Especially since this was an initial test case, I became mesmerized by the prospect of stories about knowledge propagation, which would ultimately lead to the development of *Talk of the Town*, as I explain in Chapter 9.

The Lone Mutineer

Another example of fertile initial testing of a subsystem concerns the story of the first *World* character to be marooned after a failed mutiny attempt. In this case, I saved the system's generated chronicle of her tragic life, of which the following quotation is an excerpt:[34]

> 121. Freddy Uriah is now 18 years old. Zedekiah Abbott is trying to gather up a group of people in Sidney for an expedition on the Obumbrate, which Zedekiah Abbott has agreed to lead. Freddy Uriah declined an invitation from Zedekiah Abbott to leave Sidney on the Obumbrate, for unstated reasons. Luke Uriah is trying to gather up a group of people in Sidney for an expedition on the Autolysate, which Luke Uriah has agreed to lead. Freddy Uriah hesitantly accepted an invitation from Luke Uriah to leave Sidney on the Autolysate. Freddy Uriah moved out of Dunc House in Sidney. Freddy Uriah boarded the Autolysate from the port of Sidney, Quinhagak. The group about to board the Autolysate has decided that, rather than Luke Uriah naming himself captain outright, the passengers will decide who their captain will be democratically. Luke Uriah's shipmates have decided that he would make the best captain. The Autolysate, under command of Capt. Luke Uriah, has left the harbor at Sidney, Quinhagak. The Autolysate is heading NE at half-speed. Freddy Uriah's half-brother, Capt. Zedekiah Abbott, murdered 26-year-old Jefry Archibold in self-defense at sea aboard the Obumbrate, after the latter tried to murder him. Freddy Uriah's half-brother, Capt. Zedekiah Abbott, murdered Capt. Zedekiah Abbott in an act of revenge. This was his second victim. [Note: a bug caused this character to murder himself to atone for his own killing of his half-brother]
>
> 122. The Autolysate is heading NE at full-speed. Distrust in Capt. Luke Uriah is provoking murmors of a mutiny. As an act of mutiny, Freddy Uriah is going to attempt to murder Capt. Luke Uriah, whose decision-making aboard the Autolysate has been seen by some passengers as reckless and potentially life-endangering. Freddy Uriah attempted to murder Capt. Luke Uriah as an act of mutiny, but failed. A punishment is being decided for Freddy Uriah, who acted alone in her failed mutiny attempt. The Autolysate is approaching a desolate island on which Freddy Uriah will be marooned. Freddy Uriah was marooned on an uninhabited island, which the Autolysate immediately departed. A newly discovered island is now inhabited by a company of failed mutineers marooned by Capt. Luke Uriah of the Autolysate. They call it Blighted Land. Freddy Uriah now lives alone

[34] The generation of chronicles that recount emergent events in natural language is the core functionality of *Diol/Diel/Dial*, the subject of the following chapter, and so I explain how this system works in that chapter.

in the wilderness of an island she's named Blighted Land. Freddy Uriah is questioning whether she wants to continue living after being marooned on such a desolate isle.

123. Freddy Uriah turned 20. [...] Freddy Uriah is worrying about her brother Capt. Zedekiah Abbott and mother Di Uriah, who 2 years ago both left for a voyage on the Obumbrate and haven't been heard from since.

[...]

127. Freddy Uriah decided to leave Wilderness of Blighted Land; she will do so shortly. Freddy Uriah is moving east from Wilderness of Blighted Land at a fast pace. Freddy Uriah traveled 12 miles east in the wilderness of Blighted Land.

128. Freddy Uriah is now 25 years old. Freddy Uriah decided to leave Wilderness of Blighted Land; she will do so shortly. Freddy Uriah is moving south from Wilderness of Blighted Land at a fast pace. Freddy Uriah traveled 3 miles south in the wilderness of Blighted Land.

129. Freddy Uriah decided to leave Wilderness of Blighted Land; she will do so shortly. Freddy Uriah is moving north from Wilderness of Blighted Land at a swift pace. Freddy Uriah traveled 9 miles north in the wilderness of Blighted Land.

130. Freddy Uriah decided to leave Wilderness of Blighted Land; she will do so shortly. Freddy Uriah is moving south from Wilderness of Blighted Land at a moderate pace. Freddy Uriah traveled 9 miles south in the wilderness of Blighted Land.

131. Freddy Uriah decided to leave Wilderness of Blighted Land; she will do so shortly. Freddy Uriah is moving south from Wilderness of Blighted Land at a fast pace. Freddy Uriah traveled a mile south in the wilderness of Blighted Land.

132. Freddy Uriah decided to leave Wilderness of Blighted Land; she will do so shortly. Freddy Uriah is moving east from Wilderness of Blighted Land at a fast pace. Freddy Uriah traveled 8 miles east in the wilderness of Blighted Land.

133. Freddy Uriah turned 30. Freddy Uriah is looking for a good

> site to build a house on land that she has picked out for a soon-to-be-established settlement. Freddy Uriah moved into Uriah House in Uriah. Freddy Uriah started a new settlement on Blighted Land called Uriah, which she named for herself. Freddy Uriah took her own life in Uriah, Blighted Land; she was 30 years old. Since Freddy Uriah was the only person in Uriah, she was not buried or given a gravestone.

Sentenced to Life

Another emergent story also concerns a failed mutineer, but in this case being marooned saved the character's life (and made life possible for her two daughters):

> 921. Mala Tyson turned 20. Mala Tyson reluctantly accepted an invitation from Ole Brody to leave Paint Rock on the Patesiate. Mala Tyson moved out of Freeman House in Paint Rock. Mala Tyson boarded the Patesiate from the port of Paint Rock, Zesabeta. Ole Brody has made it clear that, because it was his decision for the ship to be commandeered, that he will serve as its captain. The Patesiate, under command of Capt. Ole Brody, has left the harbor at Paint Rock, Zesabeta. [...]
>
> 922. The Patesiate is heading SE at full-speed. The crew of the Patesiate have spotted land. Capt. Ole Brody chose to pass up land and persist on the voyage, inciting a heated argument among several passengers. As an act of mutiny, Mala Tyson is going to attempt to murder Capt. Ole Brody, whose decision-making aboard the Patesiate has been seen by some passengers as reckless and potentially life-endangering. Mala Tyson attempted to murder Capt. Ole Brody as an act of mutiny, but failed. A punishment is now being decided for Mala Tyson and her conspirators in the failed mutiny attempt: Vic Thad, Lambert Thad, and Bill Mauricio II. The Patesiate is approaching a desolate island on which the conspirators of the failed mutiny attempt will be marooned. Mala Tyson, Vic Thad, Lambert Thad, and Bill Mauricio II were marooned on an uninhabited island, which the Patesiate immediately departed. A newly discovered island is now inhabited by a company of failed mutineers marooned by Capt. Ole Brody of the Patesiate. They call it Cursed Island. Mala Tyson now lives among her fellow conspirators in the wilderness of an island they've named Cursed Island.
>
> [...]

> 926. Mala Thad is now 25 years old. Mala Thad's sister, Nikolia Thad, died aboard the Patesiate, which had become lost at sea; she was 31 years old. [Note: this was the ship that marooned Mala Thad; it was lost at sea and everyone aboard died]
>
> [...]
>
> 949. Mala Thad gave birth to a girl named Nelie Bobette Thad, of whom Lonny Thad is the father. [...]
>
> [...]
>
> 952. Mala Thad gave birth to a girl named Ame Fina Thad, of whom Lambert Thad is the father.
>
> 961. Mala Thad is now 60 years old.
>
> 971. Mala Thad turned 70. Mala Thad attempted to murder Lambert Thad, but failed.
>
> 972. Mala Thad died in Kokhanok, Cursed Island, of natural causes; she was 71 years old. Mala Thad was laid to rest in the Kokhanok cemetery by a procession of its residents, who placed a gravestone there.
>
> 979. Mala Thad's daughter, Ame Thad, died in Elma, Cursed Island, after showing symptoms of disease; she was 27 years old.
>
> 1022. Mala Thad's daughter, Nelie Thad, died in Kokhanok, Cursed Island, of natural causes; she was 73 years old.

The Two Sons

In a similar tale (whose chronicle I unfortunately did not save), two sons of a prolific ship captain were aboard their father's ship when he died, thereby opening his post. One of the sons was elected captain, which offended the other one, who decided he would claim the post by mutiny. The mutineer's attempt failed, and so his brother marooned him on a desolate island; he acted alone, so there were no companions marooned with him. Shortly thereafter, the ship now captained by his brother became lost at sea, and everyone aboard died; thus, like Mala Thad in the previous story, the mutiny saved his life. On his desolate isle, the mutineer set up a port settlement, where he would live alone for the next five decades. Then, one day, a ship arrived to dock in his town, and over one hundred characters disembarked to construct houses there. These new arrivals spoke a different language, and by now the mutineer was elderly, which made learning a new language difficult. The next year, he took his life.

Emergent Poetic Images

More often, *World*'s narrative potential manifests not in full-fledged stories, but in potent situations or poetic images, like the character above who learned about his brother's murder five decades too late. In a similar image, one might encounter a character who worries about a loved one who left on a ship long ago, when in fact that ship became lost at sea shortly after setting sail, meaning the loved one is already dead. Because a ghost ship cannot be encountered (ships do not literally cross in the night), the worried character will never come to learn of the demise of that loved one. Likewise, there is the uncanny image of an elderly character slowly forgetting everyone she never knew. In this excerpt from the *Diol/Diel/Dial* entry for a character named Capt. Friedrick Prasun, we encounter both of these images at once:

> 353. Capt. Friedrick Prasun is worrying about his family member Capt. Georgianna Noland, who 13 years ago left for a voyage on the Umbelliferone and hasn't been heard from since. Friedrick is experiencing severe memory loss and no longer remembers his own daughter, Verene Prasun; they haven't seen each other since 12 years ago in in [sic] Lineville, Niota.[35]

[35] This entry is hosted at `https://users.soe.ucsc.edu/~jor/diol/5591024976.html`.

In this case, the Umbelliferone became lost at sea under the command of Capt. Georgianna Noland, and its last passengers (including the captain) died in 347.[36]

Another image is that of a character who dies alone somewhere and receives no deathbed visitors and no gravestone. The aforementioned Capt. Friedrick Prasun passed away at the improbable age of 108 as the only inhabitant in a town he had founded 63 years earlier. Others, in large settlements, attract a procession of characters who come by for one last interaction, as in this excerpt from the *Diol/Diel/Dial* entry for a character named Jan Vaclav:

> 404. Jan Vaclav has fallen ill. Jan is on his deathbed and was visited by the following friends and family: Godfree Gilles, Katleen Gilles, Benita Gilles, Clayborn Gilles, Jania Gilles, Darcy Gilles, Janus Gilles, Zia Gilles, Granville Quinn, Audrey Quinn, Pietra Quinn, Stanford Quinn, Margarethe Quinn, Jonathan Quinn, Luce Quinn, Rhodie Quinn, Teodoro Quinn, Fredrick Quinn, Carlyle Quinn, Hilary Quinn, Dosi Quinn, Paola Quinn, Thane Quinn, Winni Quinn, Waverley Quinn, Carissa Quinn, Shanna Quinn, Gabriellia Quinn, Margy Quinn, Joni Quinn, Tabby Quinn, Marsh Quinn, Malorie Quinn, Shaw Quinn, Rhodie Quinn, Yance Quinn, Horatio Quinn, Stillman Quinn, Veriee Quinn, Dori Quinn, Margy Quinn, Valdemar Quinn, Israel Quinn, Shirah Quinn, Fredrick Quinn, Holly Quinn, Rufus Quinn, Sven Quinn, Finn Quinn, Wheeler Quinn, Adrienne Quinn, Cyrille Quinn, Kinna Quinn, Sherline Quinn, Mitch Quinn, Shirah Quinn, Jerry Vaclav, Prentice Vaclav, Prentice Vaclav, Euclid Vaclav, Delores Vaclav, Armand Vaclav, Catrina Vaclav, Adair Vaclav, Gerri Vaclav, Lil Vaclav, Armand Vaclav, Avivah Vaclav, Loleta Vaclav, Cherie Vaclav, Erek Vaclav, Blondy Vaclav, Alfonse Vaclav, Edwina Vaclav, Loleta Vaclav, Marya Vaclav, Pauly Vaclav, Gunner Vaclav, Flore Vaclav, Andres Zollie, Hailey Zollie, Katya Zollie, Penrod Zollie, Julie Zollie, Luce Zollie, Katya Zollie, Juliane Zollie, Julie Zollie, Rowland Zollie, Shaw Zollie, Annadiane Zollie, Elmira Zollie, Boyce Zollie, and Spencer Zollie. Jan Vaclav died in White House, Mount Hope, of natural causes; he was 63 years old. Jan Vaclav was laid to rest in the White House cemetery by a procession of its residents, who placed a gravestone there.[37]

[36] Later, in 384, the ghost ship would finally sink. The entry for the Umbelliferone is available at `https://users.soe.ucsc.edu/~jor/diol/5591024976.html`, and the one for Capt. Georgianna Noland is here: `https://users.soe.ucsc.edu/~jor/diol/6423942160.html`.

[37] This entry is hosted at `https://users.soe.ucsc.edu/~jor/diol/6799854480.html`.

Curation and *World*

Before concluding this section, I would like to emphasize that *World* itself cannot understand or even discern the evocative stories and images that emerge in its simulated storyworlds. Even my basic presentation of these examples in this section has depended on a process of curation, in this case carried out by me. This is illustrated by my use of ellipses in the quoted passages, for example, which signals the curatorial act of removing extraneous material. In Section 8.4, I will return to these concerns in a takedown of *Diol/Diel/Dial*'s lack of curation, for which *World*'s inability to understand its own generated material is to blame.

Chapter 8

Case Study: Diol/Diel/Dial

My interactions with *World* tended to work like this: fire up a new instance, start simulating the history of its storyworld, walk away for a while, terminate the simulation procedure, and then explore the generated history (stored in working memory) by constructing the appropriate Python queries. If I was showing the system to someone else, this worked as a kind of *livecoding* [200], but I felt that I could only scratch the surface of a storyworld's emergent material, and I found myself forming the same essential queries over and over again.[1] To facilitate this kind of material excavation, I coded up a simple procedure that would take a given storyworld entity as an argument—e.g., a character or ship—and print out a brief description of this entity. The result would look something like this:

```
Name:  Kelsey Flinn
Sex:  Male
Born:  676, Wilderness of Nthyiseunt, Nthyiseunt
Died:  749 (age 73; natural causes), Flinn Creek, Minneapolis
Spouse:  Iolanthe Flinn
Parents:  Ehud Flinn and Tasia Flinn
```

[1] As I explain in Chapter 10, the notion of simulation querying as performative livecoding is central in my capacity as *Bad News* wizard. Indeed, these informal sessions of querying the emergent material of *World*'s storyworlds were probably the seed for what would evolve into the *Bad News* wizard-of-Oz [235] configuration.

```
Siblings:  Walt Flinn, Cy Flinn, Prunella Worth, Gibb Flinn,
Zondra Flinn, Quinlan Worth, Newton Flinn, Carmella Worth
Children:  Catarina Flinn, Franklin Flinn, Antonetta Flinn,
Allyce Flinn, Legra Worth, Secunda Flinn, Dominique Flinn,
Roger Flinn, Andriette Major, Ulrika Flinn, Judie Major, Carmon
Worth, Lilli Major, Sherye Flinn, Silvanus Flinn, Ola Flinn,
Isa Major
```

Though these printouts were not particularly readable, they did make it easier to investigate the emergent phenomena of a *World* storyworld. However, they only made it easier for *me* to do such investigation, since using the procedure required knowledge of the underlying codebase.

At this time in my life, my labmate Peter Mawhorter was an important mentor, and moreover a confidant with whom I was comfortable sharing my work; this was only my first year at UC Santa Cruz, and I was still new to computer programming more broadly. I had made a habit of showing him updates on the *World* project, and upon seeing my new entity descriptions, he remarked that it would be more effective if they were *hypertext*—that is, if references to other entities (such as other characters) functioned as hyperlinks that could be clicked to jump to descriptions of those entities (and so forth). This seemed like a great idea, and I set out to implement it. Initially, I sought out a means of doing this in a Python interpreter, but Peter later suggested a more general approach: generate a web page for each entity, with filenames corresponding to a unique entity identifiers, and then link between these web pages by treating the unique filenames as URIs. This was very easy to do, and within a few hours I had generated a first storyworld encyclopedia; Figure 8.1 shows a screenshot from one of these early examples.[2] Though rudimentary, these encyclopedias represented my first foray into curationist emergent narrative.

[2]Soon after, this technique would give me the idea for *GameNet* [1057], a tool for videogame discovery in the form of a network of games that are interlinked according to how related they

Kelsey Flinn

World: The World

Current year: 909

Born: 676, Wilderness of Nthyiseunt, Nthyiseunt

Died: 749 (age 73), Flinn Creek, Minneapolis

Cause of death: Natural causes

Sex: Male

Immediate family

Wife: Iolanthe Flinn

Parents: Ehud Flinn and Tasia Flinn

Siblings: Walt Flinn, Cy Flinn, Prunella Worth, Gibb Flinn, Zondra Flinn, Quinlan Worth, Newton Flinn, Carmella Worth

Children: Catarina Flinn, Franklin Flinn, Antonetta Flinn, Allyce Flinn, Legra Worth, Secunda Flinn, Dominique Flinn, Roger Flinn, Andriette Major, Ulrika Flinn, Judie Major, Carmon Worth, Lilli Major, Sherye Flinn, Silvanus Flinn, Ola Flinn, Isa Major

Figure 8.1: Screenshot from an early hypertext encyclopedia for a *World* storyworld (2014). While my typical interaction with the simulation engine was to explore the emergent material of a storyworld by querying in a Python interpreter, my labmate Peter Mawhorter suggested that I could generate a hypertext encyclopedia containing interlinked entries for all of a storyworld's entities. With the addition of a life-story field, and following substantial improvement in terms of visual presentation, this aspect of the *World* project culminated in *Diol/Diel/-Dial*, which is an encyclopedia for one particular storyworld. Though rudimentary, these early storyworld encyclopedias represented my first attempts at curationist emergent narrative.

While my first generated encyclopedias made it easier to understand a storyworld's entities and the relations among them, they did not include information

are to one another. My original prototype for the tool actually reused this encyclopedia code.

about the *events* that had emerged in that world. To capture this material, I developed the functionality that is associated with what I would now call a *chronicler* module. As I explained in Section 5.3.1, in the curationist architecture a chronicler is tasked with maintaining a *chronicle* of a storyworld, which is Hayden White's term for a raw historical record that lacks narrative structure.[3] By this functionality, which I will explain later in this chapter, the system generates a simple sentence recounting each emergent event, as it transpires, which it then appends to collections of such records that are associated with the respective entities that were involved in the event. As a result, at the end of world generation, the life of a character—or the history of another entity, such as an island or ship—may be investigated by reading a simple concatenation of all of the sentences recording all of the events of her life.

Once this system was in place, I added a field into the entity encyclopedia entries so that these prose accounts would also be included in them. When I showed this new addition to Peter Mawhorter, he remarked, "Ah, so you've got a story generator now". Until that point, I had never thought of my work as being in the realm of story generation—rather, I was intending to make a videogame—but it was interesting that a simple recounting of events seemed to have a narrative quality. This is probably where I began to think of my simulation practice as being an effort in emergent narrative.

My remaining development work on this project aimed at improving the visual presentation of the generated world encyclopedias. Here, I decided to employ a scroll skeuomorph, since that matched *World*'s vaguely historical aesthetic; Figure 8.2 shows an example screenshot. When I finished this work at the end of 2014, I had built a system that could generate a storyworld encyclopedia for any given *World* instance, including cases where a player lived out the life of a character

[3] Incidentally, in the *World* code I happened to refer to these records as 'chronicles'.

Figure 8.2: An excerpt from the *Diol/Diel/Dial* entry for Death Isle. As the entry explains, this island was first inhabited by a group of fifteen failed mutineers who were marooned as punishment by their captain, who survived the attempt. In *Diol/Diel/Dial*'s visual presentation, generated encyclopedias are stylized using a scroll skeuomorph. This entry is hosted online at `https://users.soe.ucsc.edu/~jor/diol/4524911056.html`.

through *Islanders* gameplay. In the final version of that game, an encyclopedia would be generated upon the player character's death and the player's web browser would be triggered to automatically open the encyclopedia, beginning with her character's entry. I submitted a paper on this work to the Procedural Content Generation workshop, but it was rejected. Later, in June 2015, I decided to generate an example encyclopedia and host it online as a standalone project—I called this *Diol/Diel/Dial*, since its subject storyworld was variously called 'Diol', 'Diel', and 'Dial' across its several language families. In one last attempt to make this work known, I submitted it to the Art Exhibition at the 2015 International Conference on Interactive Digital Storytelling, but that submission was also rejected. Though I found it conceptually interesting, it was not surprising to me that the work was not especially well-received—I can now articulate my sense that this was due primarily to its lack of curation, as I will explain in Section 8.4.

In this chapter, I provide an overview of *Diol/Diel/Dial*—with particular attention to the procedure for encyclopedia generation by which it was created—and situate the project against related work. Primarily, however, this chapter is a case study, as its title communicates. Below, I will provide a critique of the work in terms of the intellectual framework that I developed in Part I of this dissertation (to appraise all of emergent narrative). Specifically, I will highlight its successes in terms of the *pleasures* of emergent narrative that were articulated in Chapter 3, and then I will emphasize its failures with regard to the *pains* of emergent narrative that I identified in Chapter 4. Lastly, I will discuss this project in terms of its instantiation, or lack thereof, of *curationist emergent narrative*. This is how my subsequent case studies will be structured, as well. As a final note, while it is briefly mentioned in the paper on language modeling in *World* [1037, p. 8], in essence I am presenting *Diol/Diel/Dial* for the first time in this chapter.[4]

[4] *Diol/Diel/Dial* is hosted online at `https://users.soe.ucsc.edu/~jor/diol`.

8.1 Overview

Diol/Diel/Dial is a hypertext encyclopedia for a *World* storyworld that was generated in June 2015. A gargantuan tome of nearly 100M words, it houses biographical accounts and other information for 52,754 entities—characters, islands, wilds, settlements, ships, houses, and languages—that at one time existed in the 408-year history of the simulated world, variously called 'Diol', 'Diel', and 'Dial' across its several language families. Each character's entry contains a written record of the person's entire life, from birth to death—the longest of these is nearly 27,000 words, the length of a novella.[5] From an architectural standpoint, *Diol/Diel/Dial* is actually a collection of 52,754 individual web pages.[6] The encyclopedia is a hypertext in that each entry includes hyperlinks to several other entries. As I noted above, it is presented using a scroll skeuomorph. A reader interacts with it by navigating through its entries using conventional browser interactions: scrolling, clicking hyperlinks, and using the browser's `Back` and `Forward` buttons as needed; it does not accept interaction beyond mouse clicks and conventional browser navigation.

8.2 Encyclopedia Generation

While *Diol/Diel/Dial* recounts a specific storyworld, Diol, it was generated by a framework that is capable of producing encyclopedias for any *World* world. In each case, the data contained in a given encyclopedia is recorded as the simulation

[5] I had considered submitting the project to the National Novel Generation Month (NaNoGenMo) [573], but I never did. NaNoGenMo is an alternative to National Novel Writing Month [552] that calls for the submission of computer programs that can generate 50,000-word novels (or other text works of commensurate length).

[6] My later collaborators Matt and Chris Antoun were appalled to discover this and urged me to convert it into a more elegant web app that would dynamically generate each entry as needed (from a data store). I never found the time to do this, though.

proceeds—this is the task associated with the *chronicler* module in the curationist architecture that I presented in Chapter 5.[7] Each encyclopedia contains entries for all entities that have ever existed in its storyworld: the entire world, as well as its ocean, islands, settlements, and wildernesses; all characters that have ever lived; all ships and homes that have been constructed; and all past and present languages (along with a few prehistoric ancestors). Each individual entry, then, contains general information about its subject entity, as well as a prose account of that entity's existence in the simulation.

Entry Contents

The segment of an encyclopedia entry that lists general information about the entity includes things like a person's hometown and family members, a ship's builder, an island's discoverers, and so forth, with references to those other entities being stylized as hyperlinks to their own entries in the encyclopedia. The HTML for this portion of an encyclopedia entry is composed at the time of encyclopedia generation by querying data pertaining to these various types of information, which the chronicler tracks for each entity in the simulation.

Each encyclopedia entry also includes a prose component (under the title 'Narrative') that recounts all the storyworld's emergent events that have pertained to that entity. More specifically, this section provides a detailed yearly account of an entity's entire existence in the simulation, from its origination (or the beginning of the simulation, if it already existed then) up to the time at which the encyclopedia was generated. These prose accounts are stored as lists of strings, where each list corresponds to a year of the storyworld's history and each string recounts a

[7]In the code for this project, there is no explicitly demarcated chronicler module, but rather that functionality is carried out as part of the simulation's operation. Nonetheless, I think it is helpful to conceive of this functionality as being bundled together conceptually, and I will refer to this conceptual grouping as the 'chronicler'.

Subject	Birthplace	Template
Newborn	Settlement	[newborn.name] was born to [mother.name] and [father.name] in the town of [mother.location.name], on the island of [mother.location.location.name].
Newborn	Wilderness	[newborn.name] was born to [mother.name] and [father.name] in the wilderness of the island of [mother.location.location.name].
Newborn	Ship	[newborn.name] was born to [mother.name] and [father.name] aboard the [mother.location.name].
Parent	Any	[mother.name] gave birth to a [newborn.nominal_gender] named [newborn.name], of whom [newborn.father.name] is the father.
Sibling	Settlement	[sibling.name]'s younger [newborn.relation_to(sibling)], [newborn.name], was born in the town of [mother.location.name], on the island of [mother.location.location.name].

Table 8.1: Example templates used to produce rudimentary natural language expressions of emergent *World* events, so that they may be recounted in generated encyclopedias. For each type of event that may emerge in the simulation, I authored several template variants corresponding to different ways of recounting that event (depending on variable events details and which entity is to be treated as the subject). This was a tedious authoring task that preceded (and precipitated) my work on natural language generation.

specific event in rudimentary natural language that is written in the third person.

Event-Description Templates

The text fragments recounting emergent events are generated from templates that are associated with specific event types, such as the birth of a character or a mutiny aboard a ship. For most event types, I authored several template variants that differ according to details like which character is treated as subject or where the event took place, among other things. Table 8.1 shows a few of the many template variants that are used to generate accounts of a character's birth. The templates that I authored pertain to events of various magnitudes, from very minor events (such as a character regretting leaving their home island or worrying about a loved one who left on a ship) to very major events (such as an attempted mutiny aboard a ship). In the case of some major events, I authored multiple component templates corresponding to specific aspects of the event, which are then filled in and concatenated together to recount an entire event.

In the end, I authored templates for nearly every simulated event type, which means the chronicler is very thorough in its record keeping. Still, however, an array of simulated phenomena are not recorded, such as the deliberation processes that characters carry out in making decisions. As I have noted several times above, it is generally infeasible to perfectly capture the transpiring phenomena of a computer simulation—this is due to memory constraints, but also to the authorial burden of defining the functionality by which a chronicler may dynamically store a record of each type of phenomenon that is to be captured.

Generally, this authoring process was extremely tedious. Due to the number of event types and the number of template variants that were needed to recount each one, I was forced to craft several thousand templates. As I wrote in my rejected workshop paper on the project, the experience left me wanting more robust methods for procedural text generation (citations inserted for clarity):

> But because there were very many event types for which I authored several templates each, authorial burden in this task was still considerable; I wrote several thousand templates. [...] Moving forward, I am interested in exploring new, smarter simulationist authoring practices by which one may hand-author even smaller content units—ones corresponding to particular *aspects* of in-game events—and recombine them more smartly so that the resulting composition is more readable. Here, I might consider utilizing Tracery [204], a grammar-based authoring

> tool that handles low-level issues of recombination in a way that promotes authorial leverage. I might also consider a policy that is used in [Aaron Reed's *Almost Goodbye*] [983], in which certain language usages are kept track of and may force regeneration of text segments that reuse them too quickly.

At this time, I was just beginning to explore the prospect of natural language generation in games [1052, 1053], and within a few months I would design an authoring tool for text generation called *Expressionist* [1054], which was developed soon after with the help of several undergraduates [1049].[8] Indeed, *Expressionist*'s design was inspired by *Tracery*, and in my own projects that have utilized it, I have implemented procedures that limit repetition in the style of *Almost Goodbye*, though by a probabilistic mechanism instead. In Chapter 12, I will discuss this style of text generation as it is utilized in my project *Sheldon County*. Suffice it say that my negative experience with template-driven text generation in *Diol/Diel/Dial* precipitated my quest for more expressive, generative, and controllable methods for text generation in computational media.

Notorious People

Finally, I should note that the storyworld's entry in a generated encyclopedia (which acts as a portal to the other entries) will include a list of 'notorious people', which lists entities who are outliers with regard to various dimensions of fame or infamy. This list is constructed by querying over every character who has ever lived with regard to the kind of data that is tracked by the system's chronicler. Though not full-fledged narrativization, this is the result of curation that does involve some basic story sifting—a more satisfying curation procedure, however, would result in generated stories recounting these remarkable biographies. Here

[8] These were Taylor Owen-Milner, Max Fisher, Ethan Seither, and Tyler Brothers, the latter of whom is now leading development on the tool. We hope to release an initial version soon.

is what this component looks like in *Diol/Diel/Dial* (the entity names are stylized as hyperlinks that point to the entries for those entities):

> Lived the longest: Winifred Gilles (114 years)
> Had the most children: Reese Guillermo (24 children)
> Has the most descendants: Neville Vaclav (38065 descendants)
> Most experienced ship captain: Capt. Iain Prasun (14 years)
> Discovered the most islands: Capt. Kevin Vaclav (2 islands)
> Founded the most settlements: Ella Vaclav (3 settlements)
> Died or is living the furthest from birthplace: Anny Bert (9766 miles)
> Built the most ships: Enya Gilles (2 ships)
> Built the most houses: Aguste Vaclav (10 houses)
> Deadliest murderer: Capt. Talbert Shumeet (25 victims)[9]

8.3 The Pleasure of *Diol/Diel/Dial*

In what ways does *Diol/Diel/Dial* succeed? This is the primary consideration of this section and, as I have noted above, my discussion here will rely in large part on the ideas that were presented in Chapter 3.

Artist's Statement

For any artistic work, the most generous appraisal of the piece probably appears in its attached artist's statement, so perhaps I should start there.[10] Here is the description of the work that I included in my rejected submission to the Art Exhibition of the International Conference on Interactive Digital Storytelling, whose curatorial theme was *fragmentation*:[11]

---

[9]This information is presented in the entry for the world of Diol itself, which is available here: `https://users.soe.ucsc.edu/~jor/diol/diol.html`.

[10]To be fair, I will use these words against me in the next section.

[11]Here is the call for artworks: "Fragmentation can refer to the way the processes of remediation and transmediation are juxtaposing or contrasting stories from old or traditional media (oral storytelling, written stories, theatre, comics, radio, television, film), and how this fragmentation shapes how we experience and understand interactive stories in the new medium. At some point we need to re-fragment or de-fragment these stories so that we can make sense of new ways of telling and experiencing stories through interaction with respect to the affordances

Diol/Diel/Dial is an explorable hypertext encyclopedia of a virtual world generated by a simulationist text adventure called *Islanders*, also by the artist. A gargantuan tome of nearly 100M words, it houses biographical accounts and other information for 52,754 entities—characters, islands, wilds, settlements, ships, houses, and languages—that at one time existed in the 408-year history of the simulated world, variously called "Diol", "Diel", and "Dial" across its several language families. It evokes the old medium of the encyclopedia, but also the older medium of biography: each character's entry holds a written narrative of the person's entire life, from birth to death. The longest of these is nearly 27,000 words, the length of a novella, and each is written from an omniscient point of view that dips at times into the characters' mental and emotional worlds. In this sense, the work diverges from these old media into the computational: it could not have been made by hand (it is too massive) and could not have been made about a real world (its author, a suite of algorithms, is all-seeing and all-knowing with regard to its subject matter). While the entire volume is a perfect, totalizing account of Diol, its magnitude is such that the reader can only come to understand the world through smaller narrative fragments. A fragment, in this sense, is a reader's traversal across the hypertext—she starts at some character, clicks to read about her mother, thence the mother's birthplace, thence the ship at port there, thence its builder and his father and his language and its last speaker, and so forth. In carrying out such a traversal, the reader combines and recombines narrative fragments to form her own conception of Diol and the simulation that generated it. As such, interaction with the piece is not like exploring so much as accumulating narrative material with which to sculpt a coherent (though fragmented) view of the world. It is influenced in this way by the sculptural storymaking of Aaron Reed's *18 Cadence*, and more generally by the sheer immensity of Tarn and Zach Adams' *Dwarf Fortress*. While engaging *Diol/Diel/Dial* already feels like interactive storytelling, the piece itself is generated from a work of interactive narrative—one of the characters who lived in the simulation was controlled by a human playing *Islanders*, but unlike in player-centric game worlds, his interactions and their byproducts only ripple across the massive sum total of simulated events recounted in the larger piece. In this way, *Diol/Diel/Dial* is also an expression of the simulationist approach to interactive storytelling, and of the artist's view that non-player characters should be foregrounded ahead of human players.

---

of the new medium. Perhaps construction and sense making of these micro and mini narrative fragments also depends on the signs of time, to the zeitgeist specifically?" [315, n.p.].

Connection to Sculptural Fiction

In Aaron Reed's *18 Cadence* [981], the player sifts through an accumulation of material that captures the affairs in and of a fictional house over the course of the twentieth century; it was first described in his master's thesis [982, pp. 32–37], then later in his dissertation [984, pp. 126–130]. While it is not emergent narrative, since Reed himself authored all its materials, it is certainly curationist: players assemble fragments of the material—represented using a *magnetic poetry* skeuomorph, as Figure 8.3 shows—into lyrical recounts of what has happened in the house. This instantiates a curatorial gameplay mode that Reed identifies as *sculptural fiction* (citation inserted for clarity):

> Sculptural fiction attempts to address these shortcomings by giving the player a different kind of role. Rather than centering the act of navigation through a fixed graph of story nodes, the primary focus becomes the act of exploring the space of possible connections. Through mechanics for linking and unlinking nodes, the player builds a story, rather than moving through an existing structure. [...] The term "sculptural" is meant to suggest the constructive and iterative process of the artistic practice of sculpting, particularly the tradition of "assemblage" where existing objects are arranged to form an artistic result [979, p. 15]. The term also suggests the sculpture itself, an artifact produced by this act and designed to be exhibited and shared [984, p. 104]

This concept is clearly related to the notion of curation that I have propounded in this dissertation, particularly human-powered curation, but it differs from curationist emergent narrative in that sculptural fiction does not depend on its accumulated material being generated. As such, my framework is intended as an intervention in the area of emergent narrative, which I view as working analogously to nonfiction (see Section 3.1.1), while Reed's framework is meant to constitute a new mode of interactive fiction. In any event, at the time of creating *Diol/Diel/Dial*, I viewed the work as an effort in *generative* sculptural fiction, one

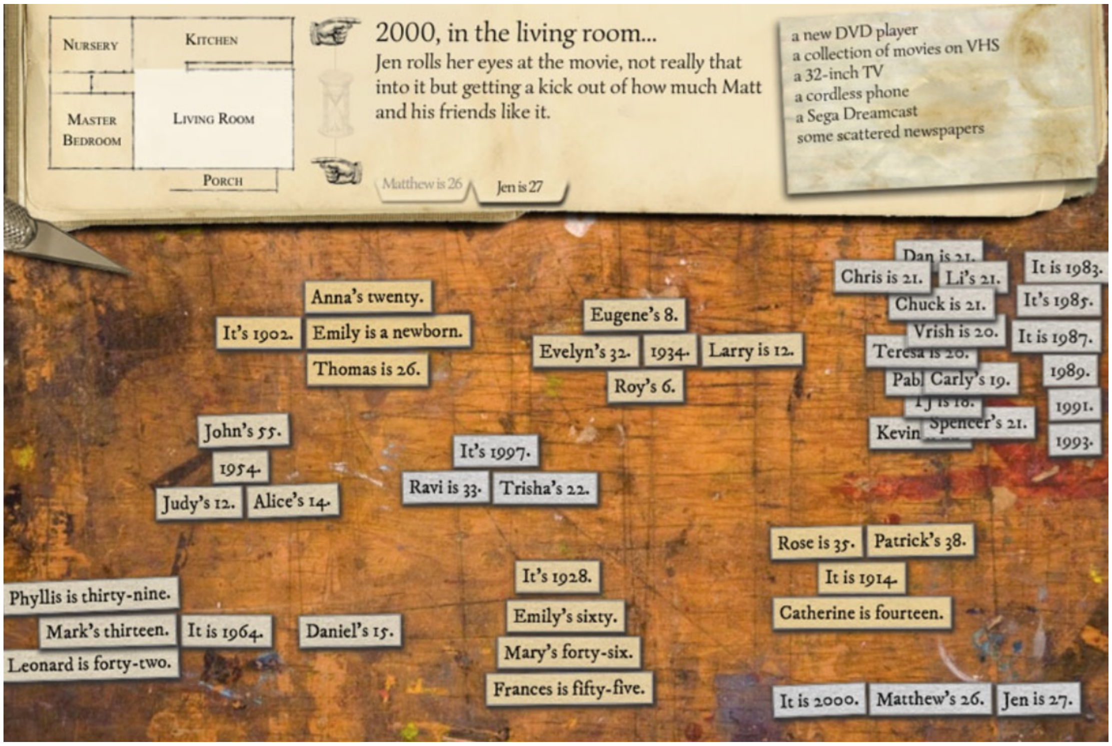


Figure 8.3: A screenshot from Aaron Reed's *18 Cadence* (2013). In this piece, the player sifts through an accumulation of material that captures the history of an American house over an entire century; in doing so, she assembles lyrical recounts of what has happened in the house. While it is not emergent narrative, since Reed himself authored all the material, it is certainly curationist. Reed articulates a particular curatorial gameplay mode called *sculptural fiction*, and at the time I viewed *Diol/Diel/Dial* as a kind of generative sculptural fiction. I now realize that such an appraisal would be overgenerous, however, due to the piece's lack of curatorial affordances.

built primarily on the pleasure of sifting.[12]

Evoking the Aesthetics of Emergent Narrative

Let us consider this project in relation to the aesthetics of emergent narrative that I identified in Section 3.2. In a sense, *Diol/Diel/Dial* exhibits the *aesthetics of the coauthored*, since a player comes to understand the world of Diol through

[12] As the next section will show, however, such an appraisal is too generous, due to the its lack of curatorial affordances, the presence of which critically structures *18 Cadence*'s gameplay.

the narrative (fragments) that she constructs about it. Moreover, in that every player will likely make a unique traversal through the encyclopedia, constructing a unique narrative understanding thereby, the *aesthetics of the personal* are likely activated to some degree.

In evoking 'the sheer immensity' of *Dwarf Fortress* in my artist's statement, I was identifying what I would now called the *aesthetics of the vast*, and there is no doubt that *Diol/Diel/Dial* is vast—it total nearly 100M words, making it an inexhaustible tome. Due to its size, each of its embedded narrative threads weaves into the massive backdrop of a record of an entire world, and in this way the *aesthetics of a larger context* are yielded. Moreover, because no human could write such a tome, the encyclopedia clearly exhibits the *aesthetics of the unauthored*. Generally, the pleasure of the work is constituted in its inclusion of the kind of emergent stories and images that I identified in Section 7.4. In that such phenomena are rare—signals in the noise—the actual emergent narrative of *Diol/Diel/Dial* has the *aesthetics of the improbable*, and the reward for sifting it out is the *aesthetics of the uncovered*. When improbability is at play, the *aesthetics of the actual* and the *aesthetics of the ephemeral* are often attendant, though in the case the latter are perhaps inhibited in that a written account persists.

Lastly, the *aesthetics of the uncanny* are clearly present in this work, if only for the ridiculously ubiquitous blood feuds that dominate the storyworld's history. This was a feature that was implemented just before generating the encyclopedia, and it is clear that the frequency of killing (and especially killing to avenge) should be tuned to be much lower. At the pinnacle of its uncanniness, *World*'s characters may even murder themselves to avenge their own killings of their own loved ones—this occurs when a character becomes obligated to avenge her victim's death. While this is a bug, its expression suggests a peculiar honor code in *World*'s procedural societies. When this is not viewed as a bug, but rather as a strange truth about these storyworlds, the aesthetics of the uncanny may obtain.[13] In Chapter 10, I will explain how a bug in the *Bad News* code that handles romantic attraction produced a peculiar aesthetic hallmark of the work.

Narrative Portals and Emergent Images

While full-fledged story sifting is not carried out in this project—as the next section argues, the task is basically impossible due to bad chronicling—an approximation appears in the form of the 'Notorious People' section (introduced above) of the entry for the world of Diol itself. This is the result of a basic curation procedure, and the links to these (in)famous entities can work as portals to potential narrative intrigue. For example, the "deadliest murderer" Capt. Talbert Shumeet is listed as having taken the lives of twenty-five victims. In viewing his encyclopedia entry, an intriguing emergent story is encountered:

> 290. Talbert Shumeet is trying to gather up a group of people in Cummington for an expedition on the Microbarograph, which Talbert Shumeet has agreed to lead. [...] Talbert Shumeet's shipmates have decided that he would make the best captain, though Dimitri Gilles is hurt by the decision. Talbert Shumeet was designated captain of the Microbarograph, which will depart Cummington shortly; he has never captained a ship before. The Microbarograph, under command of Capt. Talbert Shumeet, has left the harbor at Cummington, Waterford. The Microbarograph is heading N at quarter-speed. Capt. Talbert Shumeet married Indira Gilles on the Microbarograph; they now share quarters aboard the Microbarograph. Indira Gilles is now named Indira Shumeet. As an act of mutiny, Dimitri Gilles is going to attempt to murder Capt. Talbert Shumeet, whose decision-making aboard the Microbarograph has been seen by some passengers as reckless and potentially life-endangering. Dimitri Gilles attempted to murder Capt. Talbert Shumeet as an act of mutiny, but failed. A punishment is now being decided for Dimitri Gilles and his conspirators

[13] Of course, there a limit to this, and in any event the merits of uncanniness are highly subjective, as I discussed at length in Section 3.1.4.

> in the failed mutiny attempt: Max Noland, Hezekiah Aldis, Cassey Jordan, Uriel Prasun, Ariel Vaclav, Max Zollie, Jefferson Vaclav, Shep Gilles, Yigal Shumeet, Casandra Noland, Philis Melvin, Misha Vaclav, Jae Zollie, Hodge Gilles, Anna-Diane Zollie, Gerladina Aldis, Catarina Aldis, Pip Melvin, Emylee Aldis, Nate Vaclav, Bert Aldis, Frederica Shumeet, Breena Melvin, and Eddie Aldis. Since there are no nearby uninhabited islands on which the conspirators could be marooned, they will be hanged aboard the ship. Capt. Talbert Shumeet ordered the hanging of 26-year-old Max Noland at sea aboard the Microbarograph as capital punishment for a failed mutiny attempt in which Max tried to murder Talbert. Capt. Talbert Shumeet ordered the hanging of 23-year-old Hezekiah Aldis [...] Capt. Talbert Shumeet ordered the hanging of 44-year-old Eddie Aldis at sea aboard the Microbarograph as capital punishment for a failed mutiny attempt in which Eddie tried to murder Talbert. This was his 25th victim. [...] Capt. Talbert Shumeet is worrying about his father Capt. Addie Shumeet, who 11 years ago left for a voyage on the Microbarograph and hasn't been heard from since.
>
> 291. Capt. Talbert Shumeet is now 25 years old. The Microbarograph is heading NE at full-speed. The Microbarograph sank 275 miles off the coast of Judgement; there were 30 people aboard. Capt. Talbert Shumeet's wife, Indira Shumeet, drowned in the open ocean after the Microbarograph was shipwrecked; she was 27 years old. Capt. Talbert Shumeet drowned in the open ocean after the Microbarograph was shipwrecked; he was 25 years old.[14]

As the generated prose expresses, Shumeet's victims were in fact failed mutineers whom he ordered to be hanged as capital punishment. This scene is mildly interesting, and along with his marriage aboard the ship just prior, it forms part of an interesting year in a generated character's life. Moreover, clicking on the entry for his father, whom he worried about, reveals that the former was actually murdered by mutineers on the voyage under question.[15] The real intrigue of this story, however, is in its emergent punchline: karma shipwrecks the storyworld's

[14] This encyclopedia entry is available at: `https://users.soe.ucsc.edu/~jor/diol/5845211088.html`.

[15] The entry for Shumeet's father, Capt. Addie Shumeet, is hosted here: `https://users.soe.ucsc.edu/~jor/diol/5505259280.html`.

deadliest adjudicator off the coast of an island called Judgement. Fragments like these (and the other emergent images recounted in Section 7.4) are what provides the pleasure of *Diol/Diel/Dial*, and they are the tokens of *World*'s meager success.

It Is Mounted

Finally, I would like to note that I am satisfied with this project in the sense that I actually mounted the generated content in a full-fledged media work—there is no 'failure to mount' here, which was the curation pain that I identified in Section 4.2.5. There is a problem, however, in that a raw chronicle, rather than a set of curated narrative artifacts, is what is mounted in the experience. I will discuss this further in Section 8.5.

8.4 The Pain of *Diol/Diel/Dial*

In the style of Miguel Sicart's (2015) takedown of Miguel Sicart (2008)—[1154] vis-à-vis [1153]—in this section I will brutalize my own work in the third person (and in doing so I will traverse a strange loop).

The Aesthetic of Big Numbers Is Dead

James Ryan's *Diol/Diel/Dial* (2014) is ostensibly a work of emergent narrative, imbued with the aesthetics of emergent narrative that Ryan identifies in his dissertation [1040, pp. 92–98], but it suffers primarily for the reason that its actual emergent narrative is difficult to identify (and moreover, to isolate). In his rejected submission to the art exhibition of the 2015 International Conference on Interactive Digital Storytelling, Ryan describes the piece as "a gargantuan tome of nearly 100M words"—but as Mike Cook has declared, "the aesthetic of big numbers is dead" [470, n.p.]. It is trivial to generate massive volumes of text, and as such, the size of an artifact should never be its selling point. Yes, magnitude in this sense pertains to what Ryan calls "the aesthetics of the vast" [1040, p. 98], but such aesthetics are constituted in contextual (and conceptual) intrigue that is meant to *amplify* the appeal of an actual narrative artifact or storied experience—vastness alone is a hollow entertainment, as Cook declares eloquently.

No Curatorial Affordances

In his failed exhibition submission, Ryan seems to imply that merely traversing the encyclopedia results in a vaguely storied understanding of the material that is encountered thereby:

> While the entire volume is a perfect, totalizing account of Diol, its magnitude is such that the reader can only come to understand the world through smaller narrative fragments. A fragment, in this sense, is a reader's traversal across the hypertext—she starts at some character, clicks to read about her mother, thence the mother's birthplace, thence the ship at port there, thence its builder and his father and his language and its last speaker, and so forth. In carrying out such a traversal, the reader combines and recombines narrative fragments to form her own conception of Diol and the simulation that generated it.

Ryan then goes on to compare this experience to the sculptural fiction of Aaron Reed's *18 Cadence* [981, 982, 984], but in doing so he fails to recognize a critical component of the latter that is not present in *Diol/Diel/Dial*: Reed's interface provides *curatorial affordances* by which a player can actually assemble material into manifest narrative artifacts. In Ryan's piece, there is only jumping across links, a whirlwind tour from entry to entry to entry, the culmination of which—a mere termination, actually—leaves the player's understanding jumbled and fragmentary. To be clear, this is because she has no means with which to actually assemble a coherent understanding (let alone a narrative artifact).

Yes, there may be a kind of storied nature to the mere traversal of *Diol/Diel/-Dial*, but as Ryan himself notes in his dissertation, this is trivially characteristic of all human experience:

> When narrative psychologists write of the "storied nature of human conduct" [1093], they do not mean that human conduct itself constitutes narrative—that the raw unfolding of time is literally a story—but rather that *we can only understand it as a kind story*, since we process the world and our experiences in it through a mechanism of narrative sensemaking. [1040, p. 167]

Of course, a player who earnestly seeks to behold Diol's emergent narrative could make use of inscription technologies, such as a pen and pad, to emulate the essential experience of *18 Cadence*—but who would be so inclined? As Ryan himself notes in his dissertation, such player inclinations are typically found only in popular videogames—where they depend on attachment to a personally experienced storyworld—and indeed the more probable *dis*inclination toward curation may account for the general failure of academic emergent narrative:

> this pattern of human curation does not seem to be prevalent outside the realm of popular videogames. In the case of non-interactive story generation, and indeed many other interactive systems, humans may not be so inclined to tame the welter. To again invoke my coadvisor Michael Mateas's critique: without curation, emergent narrative is *just one damn thing after another*. [1040, p. 174]

'One damn thing after another', indeed (Mateas is my coadvisor as well). This is unfortunately an apt description of *Diol/Diel/Dial*'s 'narrative' entry components, and with all due respect to the project's creator, it is not surprising that his exhibition submission was rejected. While *18 Cadence* invites human curation, this piece makes it unpleasant work.

It Mounts a Raw Chronicle

If a work's output does not incentivize human curation of narrative artifacts, or even cultivation of a storied understanding, then the success of the project will depend on the system doing its own curation. This is the subject of Ryan's dissertation—he calls the framework 'curationist emergent narrative'—and from his description of the underlying simulation engine *World*, it is clear that such automatic curation would be infeasible in *Diol/Diel/Dial*. This is due primarily to the system's shoddy recordkeeping about past events—to use Ryan's term against him, *World*'s 'chronicler' is deficient. Intriguingly, it fails in the same way that Arthur Danto's *ideal chronicler* [239] fails: while it maintains an essentially perfect record of the storyworld *as events transpire*, such online recordkeeping does not produce narrative because the latter depends on a postmortem vantage (one that affords reasoning about causality in the large). Here, I will let Danto explain ('I.C.' stands for 'ideal chronicler'):

> Is not the I.C. definitively complete? [...] Of course it is complete—but complete in the way in which a witness might describe it, even an Ideal Witness, capable of seeing all at once everything which happens, as it happens, the way it happens. *But this is not enough.* [...] The whole truth concerning an event can only be known after, and sometimes only *long* after the event itself has taken place. And this part of the story historians alone can tell. It is not something which even the best sort of witness can know. What we deliberately neglected to equip the Ideal Chronicler with was knowledge of the future. [239, p. 154]

To be clear, in Ryan's curationist architecture [1040, pp. 232–234], a chronicler is not supposed to know the future. Instead, the 'story sifter' and 'narrativizer' modules are to operate over the chronicle after it has been constructed, with such a postmortem vantage enabling the construction of what Danto calls 'narrative sentences' [239] and what Hayden White calls 'motifs' [1332]. The problem in *World* is that the chronicler does not store records that enable the reasoning that is required to develop such narratively and rhetorically charged material. I have learned from Ryan himself that the simulation engine does not store representations of events beyond the prose strings that compose the narrative components of *Diol/Diel/Dial*'s encyclopedia entries.[16] As such, story sifting in this framework becomes a matter of natural language processing, and that is a difficult challenge.

More ideally, the system would store *structured data* about past events, which a story sifter could then reason about automatically without having to understand natural language. However, even if the system could perfectly understand the event expressions, reasoning about causal relations among them is an outstanding issue that would require commonsense knowledge about which kinds of events tend to cause others. As such, *World*'s chronicler is further deficient in its lack of 'causal bookkeeping' [1040, p. 247]. So while the pain of *Diol/Diel/Dial* is rooted primarily in its lack of curation—it mounts a raw chronicle as a media experience—this is more specifically due to poor chronicling, inasmuch as that module fails to record critical information about past events.

Horrible Prose

Beyond its poor curation, which Ryan himself acknowledges [1040, p. 359], *Diol/Diel/Dial* clearly suffers from additional presentation issues in terms of its generated prose. Here is an example passage:

> Travers Zollie is now 18 years old. Travers visited his friend Nevil Gilles, who has fallen ill and is likely to die soon. Travers visited his friend Dory Noland, who has fallen ill and is likely to die soon; they first met 18 years ago in Doyle, Waterford, where they meet now for the last time. Travers visited his friend Murphy Zollie, who has fallen ill and is likely to die soon; they first met 17 years ago in Doyle, Waterford, where they meet now for the last time. Travers visited his friend Cesya Melvin, who has fallen ill and is likely to die soon.

[16] Intrapersonal communication, June 14, 2018.

> Travers visited his friend Boyce Zollie, who has fallen ill and is likely to die soon; they first met 11 years ago in Doyle, Waterford, where they meet now for the last time. Travers visited his cousin, Armando Melvin, who has fallen ill and is likely to die soon. Travers visited his friend Friedrich Guillermo, who has fallen ill and is likely to die soon; they first met 10 years ago in Doyle, Waterford, where they meet now for the last time. Travers Zollie moved to Cummington, 11 miles away from Doyle. [...]

Ryan acknowledges this deficiency in his rejected submission to the Workshop on Procedural Content Generation:

> Even though I authored a considerable amount of content, the encyclopedias are generally not pleasing to read. For one, they are quite repetitive; this is in spite of me having written stylistic variants for most templates. More pointedly, though, they do not read well because I do not enact any discourse planning in their composition. When a new event happens, a text representation of it is composed and then appended to the existing encyclopedia representation.

As this admission points out, the poor presentation is in part rooted in the issues that I have just outlined with regard to chronicling. Discourse planning is a difficult prospect when the content of the next sentence cannot be known because it will recount an event that has not yet transpired.[17] To be fair, such *just-in-time discourse* does characterize certain genres—*sports commentary* [998] comes to mind—but this is not how discourse works in the encyclopedia genre. So while the quality of the prose does suffer due to Ryan's simple approach of template-based text generation, the real issue is that there is no reasoning at the level of discourse, since events are recounted in an online manner, as they transpire.

[17]As Stephanie Boluk and Patrick LeMieux have noted, the prose of *Dwarf Fortress*'s Legends mode [279] suffers from the same issue: "the historical sequence appears strange and arbitrary, with no transitional expressions or conjunctive adverbs to tie together the sentences" [126, p. 144]. I am not sure whether these descriptions are also generated in an online fashion; if not, poor discourse planning would instead be the cause.

An Aesthetic Hodgepodge

It is worth also noting some issues at the level of *World*'s character simulation. A primary issue in this regard is that the system's modeled domain is hodgepodge of mixed concerns whose peculiar gestalt yields a dubious aesthetics. The world is vaguely historical, yet it features the distinctly modern phenomenon of divorce; Ryan cites a Polynesian influence [1040, p. 275], but its character names are largely western-coded (and taken together constitute another strange admixture).

Simulation Pains

In terms of the simulation pains identified in Section 4.1 of Ryan's thesis, *World* does not lack. First, the system exhibits some granularity extremes, namely its low granularity with regard to its modeling of time, which proceeds by coarse year-sized chunks. While the engine is meant to generate stories that take place over the course of decades or centuries of story time [1040, p. 276], its simulation detail is low enough to significantly limit the system's possibility space. At the other extreme, the modeling of character language phenomena is conspicuously rich relative to the modeling of other phenomena—though it may represent a technical contribution to the computational modeling of language change, the subsystem is probably too detailed in that it entailed significant authoring effort while producing little narrative potential in return. Moreover, we find at least one critical modeling gap: while the simulation dips into the internal worlds of characters, the social simulation is extremely rudimentary. All of this modeling disparity violates Richard Evans's directive that "Different systems should be of similar levels of granularity" [324, n.p.].

Though the *World* does model personality, character behavior is hardly driven by such concerns. In terms of authoring, a lack of modularity has further hindered the project, as Ryan indicates in his rejected workshop submission:

> But because there were very many event types for which I authored several templates each, authorial burden in this task was still considerable; I wrote several thousand templates. [...] Moving forward, I am interested in exploring new, smarter simulationist authoring practices by which one may hand-author even smaller content units—ones corresponding to particular *aspects* of in-game events—and recombine them more smartly so that the resulting composition is more readable.

While one pitfall in emergent narrative is that all the events generated in a given storyworld may be boring—uninteresting or not *tellable* [637]—event sequences may be paradoxically boring if they are *too* interesting, as Ryan himself notes in his disseration:

> But when every story is interesting, interestingness itself is highly probable, and so the bar for intrigue is raised—this process may culminate in what Guy Bergstrom has called the *Michael Bay school of storytelling* [1040, p. 96]

In the case of *World*, and thereby *Diol/Diel/Dial*, there are simply too many murders, mutinies, and blood feuds. Of the storyworld Diol's 32,233 character deaths, 3,107 are due to murder—that is too high a rate, and the resulting onslaught (in the encyclopedia) of murder after murder makes each one unremarkable. As such, even though murder is what Roger Schank calls an 'absolute interest' [1103] and what Robert Wilensky terms a 'human dramatic situation' [1340], emergent cases of it in *Diol/Diel/Dial* are not 'reportable events', to use William Labov's phrase [632]. This does not mean that all of the storyworld's murders are uninteresting, it just means that none are *inherently* interesting; instead, intrigue depends on additional circumstances that may contextualize a murder in a remarkable way.

No Contingent Unlocking

Finally, *World*'s most acute blunder is its lack of *simulation feedback*, which limits both its narrative possibility space and the prospect of curating that space.

In his dissertation, Ryan identifies a feedback mechanism at work in Sheldon Klein's murder-mystery generator [601, 597] whereby emergent events may contingently unlock subsequent emergent events, which over time generates storylines weaved with clear causal threads:

> in this way, emergent actions in Klein's system *unlock* future emergent actions that may build upon them to grow full-fledged emergent storylines—such unlocking is an expression of the contingency relationship, and thus such emergent storylines are precisely what I have termed 'emergent contingency structures'. I call this method *contingent unlocking*. For me, the phenomenon suggests the image of a plant growing out of a spongy forest bed: meaning is constituted against the soft backdrop of entropy. [1040, p. 162]

As Ryan notes, this kind of feedback mechanism is useful not only for its potential to generate coherent emergent storylines, but to automatically *identify* them as well, by a process he calls 'causal bookkeeping'. That is, whenever an emergent event is explicitly contingent on a past event, the system records a causal relation between the two, and an emergent storyline may then be identified by reasoning over causal chains. In *World*, there are only a few kinds of events that are defined according to a contingency relation. For example, revenge killings are contingent on the earlier killings for which they exact retribution, and from this simple mechanism chains may emerge in the form of blood feuds. These are recognizable by the system—though unfortunately *Diol/Diel/Dial* does not showcase them—but the resulting sequences are not especially interesting because they are composed by a single action type. However, it should be noted that *World* does model event contingency in its mutiny simulation, as Figure 8.4 illustrates.

No Causal Bookkeeping

Unfortunately, as I have already explained, the system fails to store data about simulated events, which means that mutiny sequences cannot be reasoned about (or even retrieved) after they transpire. If event representations *were* stored in the chronicle, and if the system did do causal bookkeeping, then *Diol/Diel/Dial* could operationalize William Labov's procedure of *narrative pre-construction*:

> Before a narrative can be constructed, it must be pre-constructed by a cognitive process that begins with a decision that a given event is reportable. Pre-construction begins with this most reportable event and proceeds backwards in time to locate events that are linked causally each to the following one, a recursive process that ends with the location of the unreportable event—one that is not reportable in itself and needs no explanation. [632, p. 37]

In the case of an emergent mutiny storyline, a story sifter might begin at the mutiny attempt itself—the most reportable event—and then chain backward across causal links through the series of events that precipitated the attempt, terminating in either the captain's affront or some earlier event (pending an operational notion of tellability). Likewise, the system could chain forward in this same way, thereby retrieving the remainder of the emergent storyline. This extracted material could then be handed off to a narrativizer module that would be tasked with generating a full-fledged narrative artifact recounting the mutiny. Finally, the curated story could be mounted in *Diol/Diel/Dial*, perhaps in an entry dedicated to the mutiny itself. Moreover, if the simulation engine modeled other pockets of its narrative possibility space according to explicit contingency relations—so that the kind of structure shown in Figure 8.4 would also characterize other possible storylines—then other kinds of stories could also be curated and mounted into the encyclopedia. In this way, the experience of *Diol/Diel/Dial* would not be that of an overwhelming encounter with endless dubious prose, but one of encountering a compilation of actual stories recounting the history of a storyworld. To recapitulate, this would require both contingent unlocking and causal bookkeeping.

As he explains in the final chapters of his dissertation, the various failings of *World* and *Diol/Diel/Dial* would ultimately drive Ryan's simulation and media

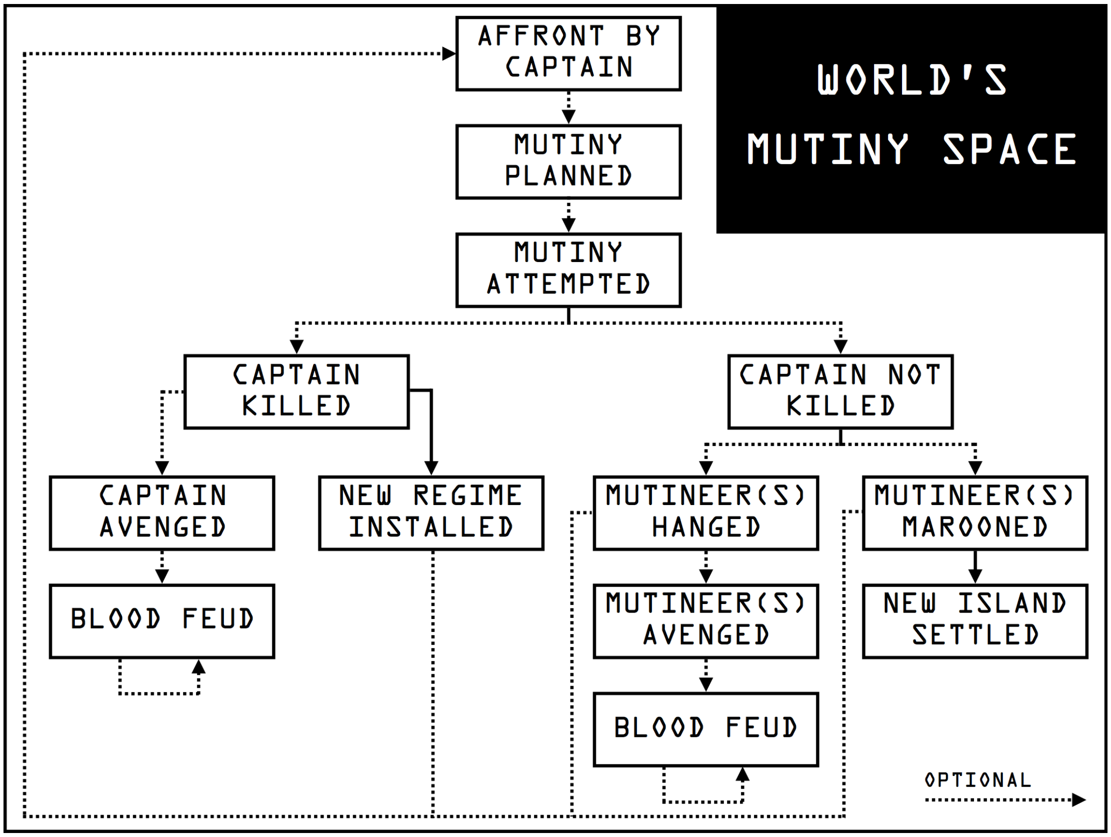


Figure 8.4: The narrative possibility space yielded by *World*'s modeling of ship mutinies. While the simulation engine generally fails to model explicit (and recognizable) contingencies between emergent events—this is perhaps the most pivotal concern in emergent narrative—its mutiny simulation is a promising exception. By its modeling, emergent events explicitly unlock the possibility of future events, thereby enabling emergent storylines with clear causal through lines (which may be automatically identified). Note that these nodes do not represent concrete plot points of the kind utilized in branching narrative, but rather authored event abstractions that may be retargeted to any emergent storyworld context.

practice toward a series of refined techniques that would eventually solidify in the form of his curationist architecture.

8.5 Curationism and *Diol/Diel/Dial*

While a series of curation pains were just articulated in the last section, briefly I would like to specifically outline how the curationist architecture, introduced in Section 5.3, is instantiated in *Diol/Diel/Dial*. As my look-alike critic noted above, this project is peculiar because it mounts a raw chronicle as a media experience. This violates the curationist call to process chronicles so that actual narrative artifacts may be constructed (through the procedures of story sifting and narrativization), which are what should actually be mounted into experiences. Curiously, while it fails to meet this call, *Diol/Diel/Dial* does go halfway in that it avoids the ubiquitous pitfall (identified in Section 4.2.1) of avoiding curation altogether to instead mount the storyworld itself as a media experience (a move typically coupled with the conflation of the raw transpiring of simulation with something that could rightly be called 'emergent narrative').

For this case study and the following, I will answer the following question: *for each curationist architectural component, who or what in this project acts as that component?* Figure 8.5 illustrates these concerns with regard to *Diol/Diel/Dial*, which I will also outline in prose, component by component:

- **Experiencer.** *Diol/Diel/Dial*'s *experiencer* is a human who visits the online encyclopedia to read through some of its entries. In the case of an encyclopedia being generated from an *Islanders* storyworld, there is another experiencer who interacts with the simulated storyworld that is itself recounted in the encyclopedia. In this case, the experiencer of the encyclopedia may be the same person as the *Islanders* player, or she could be someone else. Figure 8.5 does not capture the case of *Islanders* gameplay.

- **Simulated storyworld.** A *World* simulation instance (the one in which

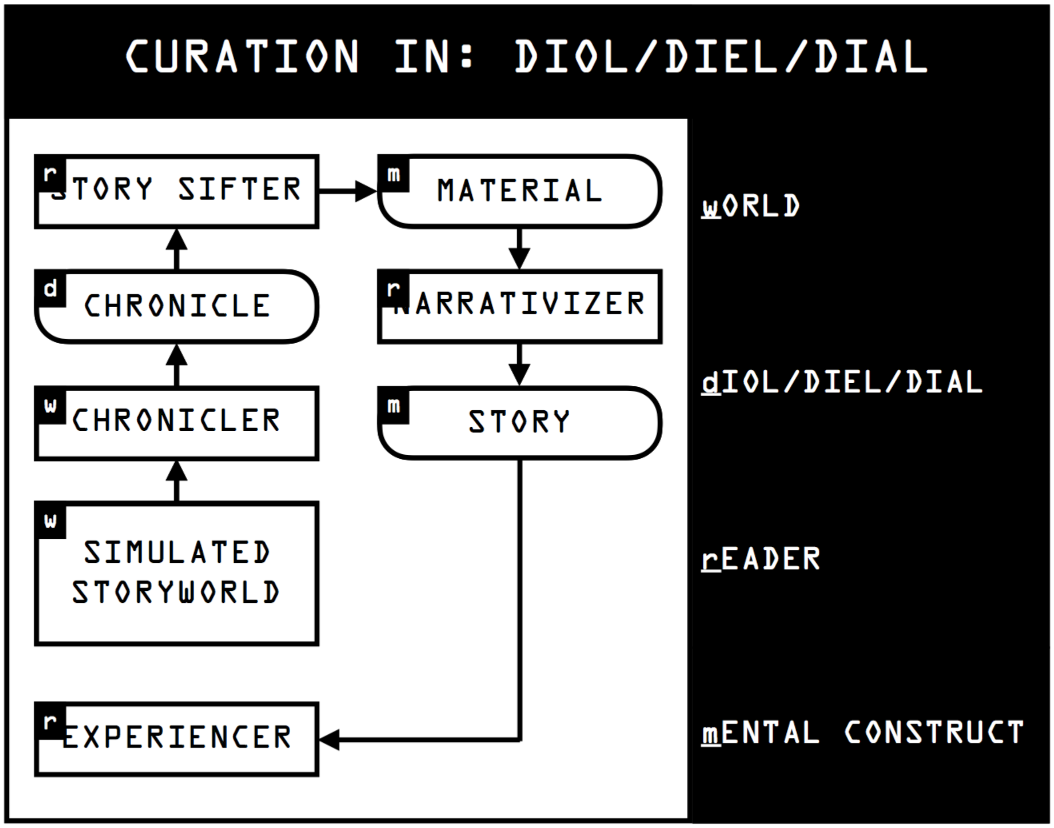


Figure 8.5: Curationist emergent narrative in *Diol/Diel/Dial*. In this diagram, letter designators placed in the corner of each architectural component cue the corresponding entities in the *Diol/Diel/Dial* project. Here, we find a variation on the *mental curation* pattern shown above in Figure 5.4: the transpiring events of a *World* storyworld are recorded in a chronicle, which is itself mounted as the *Diol/Diel/Dial* experience. As in other experiences that entail mental curation, a reader must carry out the actual curation procedure by which emergent stories may be constructed (in the reader's mind). More typically, mental curation characterizes experiences in which the simulated storyworld is itself interactive, which makes curation a process of mentally narrating on ongoing experience with a simulation. In this case, we find something more akin to historiography, where a reader encounters a chronicle that recounts events in which she was not a participant. I view this project as a failure primarily for its lack of curation—its blunder is specifically that it mounts a raw chronicle as a media experience—but my eventual realization of that failure led to an increased emphasis on curation in my simulation practice, which Figures 10.9 and 12.1 will illustrate.

Diol obtained) is the *simulated storyworld*.

- **Chronicler.** As I noted above, there is no distinct chronicle module at work in *Diol/Diel/Dial*, but rather the associated functionality is built into the *World* code. To simplify, we can say that *World* is its own *chronicler*.

- **Chronicle.** A *chronicle* in this project comprises all of the various data that is included in the generated encyclopedia; in particular, the narrative components of its entries (concatenated string representations of events) most resemble Hayden White's notion of a chronicle. As such, we can say that *Diol/Diel/Dial* is itself a chronicle, and curiously one that has been mounted as a media experience. *Dwarf Fortress*'s Legends mode [279], likely the most similar existing work to *Diol/Diel/Dial*, has itself been likened to a Whitean chronicle [126, pp. 143–149].

- **Story sifter.** Here is where the curation pain begins—in *Diol/Diel/Dial*, the human encyclopedia reader is herself tasked with carrying out the duties of the *story sifter*. This is because the system does not do this automatically, which is in turn due to the simulation not being crafted in a way that would facilitate automatic sifting (see Section 8.4).

- **Material.** In this project, the sifted *material* is simply a mental construct in the mind of the human encyclopedia reader. As she traverses through a series of entries, knowledge capturing the information she has encountered so far accumulates in her mind, and in total this is the sifted material. Alternatively, if the reader brings inscription technologies to bear—for instance, by taking notes using a pen and pad—then the material also obtains in the form of her inscriptions.

- **Narrativizer.** The human encyclopedia reader is also forced to carry out the duties of the *narrativizer*. If she comes to discern emergent storylines in the materials that she has excavated—be they mental constructs, inscriptions, or a combination—that she has performed narrativization.

- **Story.** If the human reader does indeed perform narrativization, she produces a narrative artifact that recounts Diol's emergent phenomena. Specifically, these phenomena will have transpired in Diol before being captured in the chronicle (*Diol/Diel/Dial* itself) and then sifted out of the chronicle in the form of material, which will have finally been narrativized into a *story*. This story is likely another mental construct that obtains in the mind of the reader, in which case *mental curation* (see Section 5.3.2) has occurred. Of course, she could also render the story in another form by mounting it in a media experience in the style of Tim Denee's *Oilfurnace*, in which case *feed-forward curation* would be at work. The other architectural variant, *feedback curation*, is not possible here because it would require the reader to inject her constructed story back into the storyworld of Diol, but she has no means to do so. (In fact, Diol is gone forever and can never be retrieved—unlike *Talk of the Town* and *Hennepin*, storyworlds do not have associated seeds, which means they cannot be recovered once a *World* session ends.)

- **Media experience.** *Diol/Diel/Dial* is of course the *media* experience in this project, but again it is curiously also the chronicle component. To reiterate the takeaway of this chapter, the mounting of a raw chronicle as a media experience is a major blunder. Also, as I have just noted, it would be possible for a reader to curate the emergent material captured in the chronicle (in *Diol/Diel/Dial* itself) to construct stories that could be mounted in a second-order media experience. To my knowledge, no one has done this.

Summary

To summarize, *Diol/Diel/Dial* employs the degenerate architectural variant of *mental curation* (see Section 5.3.2), since it leaves story sifting and narrativization as exercises for the reader. However, while in most examples of mental curation the storyworld is not curated at all, in this case a chronicle is automatically produced (and then mounted as the media experience, rather than the storyworld itself). While *feedforward curation* is possible here, it would require the reader to mount her curated stories about Diol into a second-order media experience, and to my knowledge this has not occurred. As I will discuss in the next case study, *Bad News* represents a novel experiment in having the experience itself do its own curation, but through the mechanism of a wizard-of-Oz hybrid human–computer scheme. Later, in Chapter 12, I outline my ongoing exploration of fully automatic curation in the generative podcast *Sheldon County*.

Chapter 9

Talk of the Town

At the beginning of 2015, during my second year at UC Santa Cruz, I took Noah Wardrip-Fruin's *Playable Media* class [1314] for a second time. As I mentioned in Chapter 7, this course requires students to distribute a project to the rest of the class for critique; in the previous instance I had developed *Islanders*, the text adventure built atop the *World* simulation engine. This time around, Adam Summerville—a newly arrived PhD student in the Expressive Intelligence Studio who was also taking the class—asked if I would like to work together with him to distribute a collaborative project for critique. I agreed, and offered up some initial ideas that had been brewing in my mind.

My first idea was to build a rhetorically charged inversion of *Super Mario Bros.* [879], which I described in this way in an email to Adam:

> A networked multiplayer (sub)version of Super Mario Bros., where Mario is the AI and humans control various NPCs in a level (with their exact affordances intact) and collaborate to try to stop Mario. If I did this, I was planning to call it Super Bowser Corps and make a new manual for the game, where the subverted narrative is described: Bowser is this heroic revolutionary who's toppled the hegemonic Mushroom Kingdom in a liberation effort involving persons of all kinds (i.e., the Mushroom Kingdom supports only toadstools, but Bowser's minions come from several species, or whatever you'd call them), but now

> there's a pair of evil plumbers who are undermining the heroic effort for purely personal reasons.[1]

As Adam pointed out, the idea was probably too ambitious for a class project, since it would require implementing multiplayer support for real-time gameplay. We decided not to go that route, though it is interesting to ponder about now, since it would have represented a major departure from the kind of work that I had done to that point and since.[2] Another idea that I proposed was more directly in line with the simulation work I had been doing on *World*:

> A game where knowledge propagation is the core mechanic. I had two ideas for premises. The first is a multiplayer game where two players are both chasing each other, and their primary means for tracking is knowledge acquisition by dialogue interaction with NPCs [non-player characters]. The other premise is an asymmetric multiplayer game where one player is on the run and attempting to reach a certain location before another character, who's managing a network of NPCs over which knowledge is propagating, tracks them down. In both of these, the world would have a fairly rich simulation to allow for interesting knowledge prop patterns.[3]

At this point in my simulation practice, I had become particularly interested in two new directions: character knowledge propagation and the prospect of automatically recognizing interesting emergent stories. While the latter would gradually develop into an increasing emphasis on curation, itself the subject of this dissertation, I had already begun to explore the former notion in *World* and I was very excited about the possibilities. As I recounted in Chapter 7, knowledge phenomena in that system is extremely limited—characters exchange last known whereabouts for common acquaintances—but it had captured my imagination

[1] Personal communication, January 23, 2015.

[2] Adam, on the other hand, would go on to do a lot of work in the context of the *Super Mario* series [1219, 1218, 1214, 1215, 1213], though in the area of procedural content generation. My idea was of course incidental to his interest in the series.

[3] Personal communication, January 23, 2015.

since an initial test in which an elderly man was reunited after sixty years with a childhood friend, who proceeded to tell him, one by one, when and where everyone he had known had died in the intervening years. This emergent phenomena was enabled by just a few days of work, and I wanted to explore the topic more deeply in a successor system.[4] In his response to my email, Adam expressed interest in the idea and proposed that we could even model interesting memory effects:

> This is the one that is the most intriguing. [...] I'm a huge fan of asymmetric multiplayer, and I feel like there's a lot that could be done here. The obvious things are the player trying to disseminate false information about who they are and their intentions, and classic memory degradation for witnesses as well as priming (both of which I think would be very interesting effects to model).[5]

From here, we set out to design a game that would be *about* character knowledge phenomena—later on, in an email describing the project to our labmate Ben Samuel, I wrote: "The idea is to deal in character knowledge as richly as Prom Week deals in character social interactions, so 'knowledge physics', if you will".[6] We decided that adapting the *World* simulation technology would give us a head start on the project, but we felt that a contemporary urban setting, where knowledge propagates quickly, would work better than island archipelagoes. Eventually, we realized that it would not be feasible to model an entire city filled with *World* characters, so we converged on the American (very) small town as our domain—something on the order of a few hundred characters. In the end, I took this as an opportunity to abandon the *World* codebase—which was my first major programming project, and thus of dubious structural and aesthetic quality—and to do an intellectual reboot in terms of my approach to modeling.

[4] Apparently, I was initially interested in doing a project for the class that would also extend *World*'s language simulation, as I jotted in a short note at that time: "Knowledge propagation in a language-contact zone—speakers of different languages have a pidgin, but knowledge propagation is impoverished in this mode. Aesthetics: flat design; dull pink and lime green".

[5] Personal communication, January 23, 2015.

[6] Personal communication, August 13, 2015.

As for the actual game, much of our early design work centered on the construction of a narrative premise that would foreground character knowledge and support our vision for asymmetric multiplayer. The obvious framing was gritty crime and detection, but we wanted to explore territory that would be thematically peculiar for a videogame—our primary influence was Wes Anderson's *The Grand Budapest Hotel* [43]. Here is our description of the narrative premise that we eventually invented, as published in a first paper on the project:

> The story that frames gameplay surrounds the death of a very important person who, seventy years prior, founded the town in which gameplay takes place. Since that time, the founder has accumulated considerable wealth and produced several descendants who now constitute an aristocracy in the town. Many of these family members had been anticipating the founder's death for the inevitably large inheritances that would thereby be disbursed, but in his or her last moments the founder apparently signed a document willing everything to a secret lover whose existence had not been known to the family. In one week, the town will gather at a theater to remember the deceased and to hear the will be read, but the family plans to ascertain the identity of the lover and apprehend this person before the document can ever be delivered to the presiding attorney. Meanwhile, the town is abuzz with rumors about the mysterious lover, whom a handful of witnesses briefly observed on the night of the founder's death. [1061, p. 57]

We felt that this framing would support our targeted pattern of asymmetric gameplay centered on character knowledge phenomena, which we described in this way:

> The game is multiplayer and asymmetric: one player controls the lover character and the other player controls a member of the founder's family. The lover's goal is to go undetected until the will ceremony, while the family member works to ascertain the lover's appearance before that time. [...] Because the family character is established in the town, the player controlling him or her will have the town's entire knowledge network at her disposal. As such, her job becomes *managing* this network so that information about the lover's appearance flows toward her; the lover player's task then is to *pollute* this knowledge network by, for instance, changing the character's appearance and spreading lies. Gameplay culminates in a scene showing the town's citizens filing

> into the theater for the will ceremony, during which time the family player must select the person who best matches her conception of the lover—if she selects correctly, she wins; otherwise, the lover player wins. [1061, p. 57]

In line with my proclivity for world generation, already established in *World*, we had always planned for the town to be procedurally generated prior to gameplay. Critically, this procedure would entail the simulation of its entire history, during which time characters would form and propagate knowledge about one another. During gameplay, the town would be represented as a 3D space that players could navigate using the visual metaphor of a board game (from an aerial view, click to select destination). The core gameplay experience was to be conversational interaction with characters in town—this is the context in which players would be able to ask questions (inquire about knowledge) and also make statements and tell lies (pollute knowledge).

Though we never completed the game, we did build a prototype that was distributed to the class; Figure 9.1 shows screenshots from this software. Due to a limited development period (six weeks), we could not produce a multiplayer experience, but instead a single-player configuration in which the player attempted to discover the name of the lover. As with my *Islanders* project, which was developed for the same class the previous year, the critique did not go particularly well. We faced a number of technical challenges in developing the game—it was a Unity project [405] and integrating my Python simulation engine became a nightmare for Adam—and due to the amount of character knowledge that players could inquire about, conversational interaction required navigating an unwieldy series of nested menus. There was a lot intrigue emerging in the simulation, but significant design and technical challenges led to that intrigue being obscured at the surface—we were up against the *Tale-Spin effect* [1310, 1311].

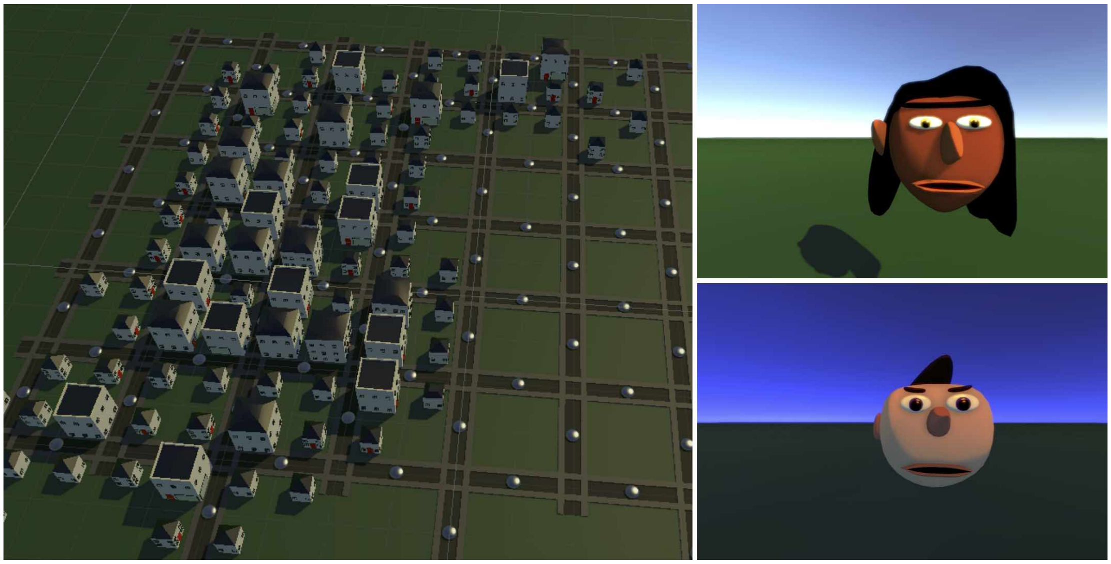


Figure 9.1: Screenshots from the videogame prototype *Talk of the Town* (2015), which Adam Summerville and I developed as a class project. Our goal was to produce an asymmetric multiplayer game centered on character knowledge propagation and conversational interaction. Though we never completed the game, its underlying simulation engine found new life as a system confusingly also named *Talk of the Town*. Many of the concepts that originated in our design for this game were ultimately realized in the completed experience *Bad News*, the subject of Chapter 10, for which Ben Samuel joined Adam and me as a third codesigner.

While eventually we abandoned development on the game, many of the concepts that originated in its design were ultimately realized in the completed project *Bad News*—I will tell that story in the next chapter. Through that work and other experiments, the simulation engine that drove the *Talk of the Town* prototype took on a life of its own. Confusingly, however, I also refer to this simulation engine as *Talk of the Town*. This is because the moniker had already become associated with the simulation through early papers on the project that were primarily about that component [1061, 1043, 1044].[7] As I mentioned in Chapter 7, a similar confusion had transpired with regard to the titles *World* and *Islanders*, and generally

[7]From this point forward, I will refer to the simulation engine as *Talk of the Town* and to the videogame prototype as *Talk of the Town* (2015).

this was a manifestation of conceptual murkiness in my simulation practice. In wrangling the notion of what exactly 'Talk of the Town' had come to denote, I began to realize what that practice is all about: building simulation *engines* that are themselves standalone media works, but ones that are ultimately meant to drive *second-order media experiences* like *Talk of the Town* (2015) or *Bad News* or the other examples that I will mention later in this thesis.

Talk of the Town is a simulation engine that models an American small town over the course of its history, with particular attention to the lives of its denizens. Its modeling of social phenomena is fairly abstract, though it supports some interesting emergent dynamics; its hallmark is a rich modeling of character knowledge phenomena, particularly memory fallibility. As I have noted above, *Talk of the Town* was initially created in the context of a collaboration with Adam Summerville, and much of the core details are the result of design brainstorming that we conducted at that time. I continued to develop the engine through the beginning of 2017, at which point I abandoned it to begin work on its successor, *Hennepin*, which is the subject of Chapter 11. While I have carried out a number of (collaborative) experiments with the engine that led to the introduction of experimental modules—notably ones pertaining to conversational interaction [1045, 1059, 1220, 1050, 1046] and character streams of consciousness [1042]—I will describe it in this chapter in terms of the specific instantiation that is used to drive *Bad News*, since that is the case study of the next chapter.

Again, a note on evaluation: I believe that computational systems that are intended to support media experiences can only be appraised through actual implemented experiences that are built atop them, as others have articulated before me [762, 1200, 505, 610, 1316]. Because the purpose of *Talk of the Town* is to drive second-order media experiences, I will hold off on appraising it in that regard until the following chapter, in which I will critique it through the lens of *Bad News* (an actual implemented media experience built atop it). In lieu of such appraisal, in this chapter I will instead situate *Talk of the Town* against related technical work, both in a broad sense of its approach to simulation and also in terms of some of its specific subsystems, particularly the one that models character knowledge phenomena. For clarity, I discuss this related work in a distributed manner, throughout the chapter, as the pertinent concepts arise in my discussion. While specific aspects of the system have been reported in earlier papers, this chapter will provide the first published explanation of its general machinery for building and simulating American small towns.[8]

9.1 Modeling

Each *Talk of the Town* storyworld is modeled as a small town that is populated by a few hundred characters, who construct houses and other buildings in the town, some of which headquarter business operations. In this section, I will give a basic overview of the modeling of characters and the various entities that are included in the simulation. As I explained in the opening of this chapter, this simulation engine was originally developed to drive a videogame with a particular design. This is the context in which the majority of design decisions were made, and so a few peculiar aspects—namely the overwrought modeling of character appearances—are artifacts that remain from this origin. Finally, unlike *World*'s hodgepodge domain, *Talk of the Town*'s is explicitly the American small town.

[8] In the case of subsystems that have been reported before, I will note the earlier publications as I describe those subsystems. Though it has not appeared in print, I have provided an explanation of how towns are generated and simulated in two invited presentations: "Generating American Small Towns for Narrative Applications", delivered at the 1st Workshop on Tutorials in Intelligent Narrative Technologies, held in Los Angeles on November 15th, 2016; and "How to Make an American Small Town", given at the University of Turin on October 9th, 2017.

This is as specific as it gets, however—it could be a small town in any area of the country (though certain business types, such as a coal mine, may suggest something about the region of a particular town instance).[9]

9.1.1 Time

Talk of the Town models time more finely than its predecessor *World*: whereas in the latter engine time passes one year at a time, in the former it proceeds by *day* and *night* timesteps. Thus, the simulation deals in roughly half-day chunks, marching from day to night to day to night and so forth.

For *Bad News*, however, I do not simulate every timestep in the history of a town at full granularity, but instead utilize *level-of-detail modulation*.[10] Level-of-detail modulation is a general technique whereby the granularity of simulation or representation is adjusted for some system according to various contexts, usually under the influence of computational constraints. While it originated in computer graphics [711], the concept has also been applied to character simulations [270, 741, 892]. In the case of modeling a *Bad News* storyworld, *Talk of the Town* simulates approximately four days each story year, and then critically extrapolates the results of character social simulation on a given timestep according to the amount of time that has passed since the last simulated timestep. For instance, if 200 timesteps have passed since a timestep was last simulated, and two characters

[9]Some observers from coastal regions have assumed that the towns must be based in the Midwest. I am bothered by that assumption, since it evokes the problematic stereotype that the Midwest is a miscellanous area dotted with farming communities and small towns. Every region in this country has both urban and rural areas, and the threshold between those is the nation's most significant cultural divide—not coastal versus interior. The region called the 'Midwest' is a meaningless conglomeration of distinct regions that have little in common, and its very reification reflects a troubling coastal bias. Minnesota has more in common with Massachusetts than it does with Missouri, for example. I refer to my hometown of Minneapolis as being in a region called the 'North', following a movement that is afoot to recast it as such [109, 980, 969]. This rant is courtesy of a Minnesotan's five-year stint in California.

[10]Again, the purpose of this chapter is to describe *Talk of the Town* as it is utilized in *Bad News*, since that is the case study of the next chapter.

engage in a social interaction whereby one comes to like the other more by some unit of increase, her affinity for that character will be increased by a level corresponding to 200 of those units. This makes individual social interactions between characters stand for a period of game months characterized by extended interaction between the dyad. This saves on computation, and critically I have found that this measure does not compromise the richness and coherence of generated storyworld material.[11] I will return to this consideration below, in Section 9.2.5.

9.1.2 Characters

Characters have been the central concern in my simulation practice ever since I developed the name generator described in Chapter 6, and this project is no exclusion. In this section, I will describe what a *Talk of the Town* character is made of. In doing so, I will emphasize what this engine inherits from *World*, as well as how it departs from its predecessor. Again, a core aim of Part II of this thesis is to describe the evolution of my simulation practice over time.

Name

The procedure for giving characters *names* in *Talk of the Town* generally follows that of *World*, but with some revisions that reflect the former's specific domain of the American small town. When a new character is born, she may be named for a family member (at a set probability), but generally characters are named according to census data that captures the most common baby names throughout American history. If a character is to be named after a family member, a probabilistic procedure determines whom that will be with likelihoods depending

[11] As I will explain in the next chapter, each *Bad News* storyworld is generated just as a player is arriving to our installation to experience the next performance. In the interest of scheduling as many performances as possible, we need to limit the amount of time it takes to produce a storyworld. This is the basic impetus for doing level-of-detail modulation in this way.

on the relation of the infant to a set of potential namesakes. Otherwise, naming works like this: for the character's year of birth, look up the relative frequencies of the 200 most common baby names for that year, and name the child accordingly (by deriving a probability distribution from the relative frequencies). This leads to names like 'Mary' and 'Elmer' giving way to examples like 'Deborah' and 'Kevin' and eventually to 'Ashley' and 'Ryan' and so forth.[12] Additionally, there is a small chance that a character will be named by randomly selecting from the forenames corpus that I have been using since my early name generator. Deciding a character's middle name works in this same way.

Character surnames come from the same surnames corpus that was utilized in *World*, and following American custom, a patronymic naming scheme is enacted. However, parents may also decide to give their children hyphenated surnames. This occurs at a set chance in the codebase that drives *Bad News*, but later I altered it to change over time according to attested data—this is the modeling used in *Hennepin*, as I will explain in Chapter 11. As in *World*, a character with the same full name of a parent is given the appropriate numeral suffix, and the system tracks the namesakes and ultimate originating character for each forename and surname. Finally, the system keeps track of the ethnicity of surnames, which is discernible due to the surnames being grouped in the ethnically delineated corpus. As I will explain in Section 9.2.12, characters may reason about this information and a surname's ethnicity affects how it may be misremembered.

[12]'Ryan', which is my surname, was essentially unused as a forename (in the United States) prior to the 1940s, and even then it was obscure until the 1960s, when its popularity exploded [59]. When I would introduce him to friends with that name, my late grandfather, Jack Ryan, reacted with incredulity: "Ryan!? That's a last name!"

Appearance

As I explained in this chapter's opening, the game design that Adam Summerville and I devised for *Talk of the Town* (2015) was in large part about character *appearances*: the family member attempts to ascertain the appearance of the secret lover so that she may pick her out of a lineup at the will ceremony in which gameplay culminates.[13] Due to its importance in this design, *Talk of the Town* richly models character appearance, to a degree that may seem overwrought with regard to its more abstract modeling of other physical concerns. I should also note that the introduction of character appearance represents a fundamental addition relative to *World*, which did not model any physical characteristics of characters beyond what is implied by one's age.

What follows are the system's twenty-four appearance attributes (which pertain primarily to the face) and the values that characters may take for each:

- Skin color: `black`, `brown`, `beige`, `pink`, `white`
- Head size: `small`, `medium`, `large`
- Head shape: `square`, `circle`, `heart`, `oval`
- Hair length: `bald`, `short`, `medium`, `long`
- Hair color: `black`, `brown`, `blond(e)`, `red`, `gray`, `white`, `green`, `blue`, `purple`
- Eyebrow size: `small`, `medium`, `large`, `unibrow`
- Eyebrow color: `black`, `brown`, `blond(e)`, `red`, `gray`, `white`, `green`, `blue`, `purple`
- Eye size: `small`, `medium`, `large`
- Eye shape: `round`, `almond`, `thin`
- Eye color: `black`, `brown`, `blue`, `green`, `yellow`
- Eye horizontal settedness: `narrow`, `middle`, `wide`

[13]The subsystem discussed in this section was designed in collaboration with Adam Summerville, in the context of development of *Talk of the Town* (2015). Throughout Part **??** of this thesis, in which I report both independent and collaborative work, I will attempt to clearly attribute my collaborators and their contributions.

- Eye vertical settedness: `high`, `middle`, `low`
- Ear size: `small`, `medium`, `large`
- Ear angle: `flat`, `protruding`
- Nose size: `small`, `medium`, `large`
- Nose shape: `long`, `broad`, `upturned`, `pointy`
- Mouth size: `small`, `medium`, `large`
- Facial-hair style: `none`, `mustache`, `full beard`, `goatee`, `sideburns`, `soul patch`
- Freckles: `no`, `yes`
- Birthmark: `no`, `yes`
- Scar: `no`, `yes`
- Tattoo: `no`, `yes`
- Glasses: `no`, `yes`
- Sunglasses: `no`, `yes`

Heritability with regard to character appearance is modeled using a similar procedure to the one that drives the inheritance of personality traits in *World*, which I described in Section 7.2.2. When a child is born, the system consults a set of heritability probabilities that I authored for each trait to determine whether a given trait will be inherited or not. If inheritance is triggered, one of the child's parents is randomly chosen and that character's trait value is inherited. If inheritance is not triggered, the system will generate a value for the character using a set of population distributions that I authored for each trait—this loosely models a kind of genetic mutation (or latent features recurring after skipping a generation). Some traits are not heritable (e.g., whether a character wears sunglasses), and a child's skin color is set at a midpoint between her parents' respective tones. Of course, characters without parents—*primordial* characters who already exist at the beginning of the simulation (about which more below)—cannot inherit trait values. Finally, a character's hair color changes over time (by probabilistic triggering), first to `gray` and eventually to `white`.

Personality

Just as in *World*, character *personality* in *Talk of the Town* is driven by the famous *five-factor model of personality* [810, 1337], which is also referred to as the *Big Five* model. This subsystem works exactly as it does in my earlier engine, which means I have already described it at length in Section 7.2.2. To briefly recapitulate, the personality traits in this model are *openness to experience*, *conscientiousness*, *extroversion*, *agreeableness*, and *neuroticism*. Values for each trait are represented as floating-point numbers on a continuous scale from `-1.0` to `1.0`. Children inherit these values from their parents, with a mutation factor that modulates the exact value instantiated in the child. Character personalities are static profiles that do not change over time. As I have alluded to above and will explain in Section 11.1.2, in *Hennepin* I am no longer using the five-factor model—this is because I have come to believe that the model is not dramatically potent and not particularly conducive to authoring character behaviors.

Knowledge

A character's accumulated *knowledge* of the storyworld is structured as a collection of mental models, each of which is comprises a variety of beliefs about an individual entity in that world. Section 9.2.12 is devoted to the subsystem that handles character knowledge phenomena in *Talk of the Town*, which is the system's core technical contribution.

Miscellaneous Attributes

Talk of the Town characters are also modeled according to several additional attributes beyond name, appearance, personality, and knowledge. These include *sex*, *sexuality*, *fertility*, and *memory*, the latter of which is modeled as a floating-point value that decreases over time to make the phenomena pertaining to memory fallibility more likely. As in *World*, a character's sexuality is composed according to whether she is probabilistically determined to be attracted to men and/or women. Additionally, the system tracks character-specific data pertaining to a number of concerns including social relationships, romantic history, work history, residential history, and more.

9.1.3 Other Entities

In addition to characters, *Talk of the Town* models the town, of course, including its streets, tracts, lots, businesses, homes, and a few artifacts. I will describe each of these in this section.

Towns

A town is represented as a rectangle that spans nine city blocks on each side. Before the town is created, a plat is generated for it using a *quadtree* [1081, 1132]. This procedure was developed by Adam Summerville and works as follows: subdivide a rectangle (spanning nine by nine city blocks) into four squares, probabilistically subdivide those squares into four smaller squares, and repeat for a number of iterations. The result is a grid of variably sized rectangles, but the procedure is tuned so that nearly all the rectangles will be squares spanning one city block on each side. Unlike *World*'s settlements that grow over time and annex one another, described in Section 7.3.9, the layout of a *Talk of the Town* town does not change over time (except that roadways are constructed as needed). I view this as a limitation, since character decisions (and disagreements) about how to change the town would yield intrigue and narrative potential. As I explain in Chapter 11, *Hennepin*'s counties (and their component towns) are dynamically reconstructed

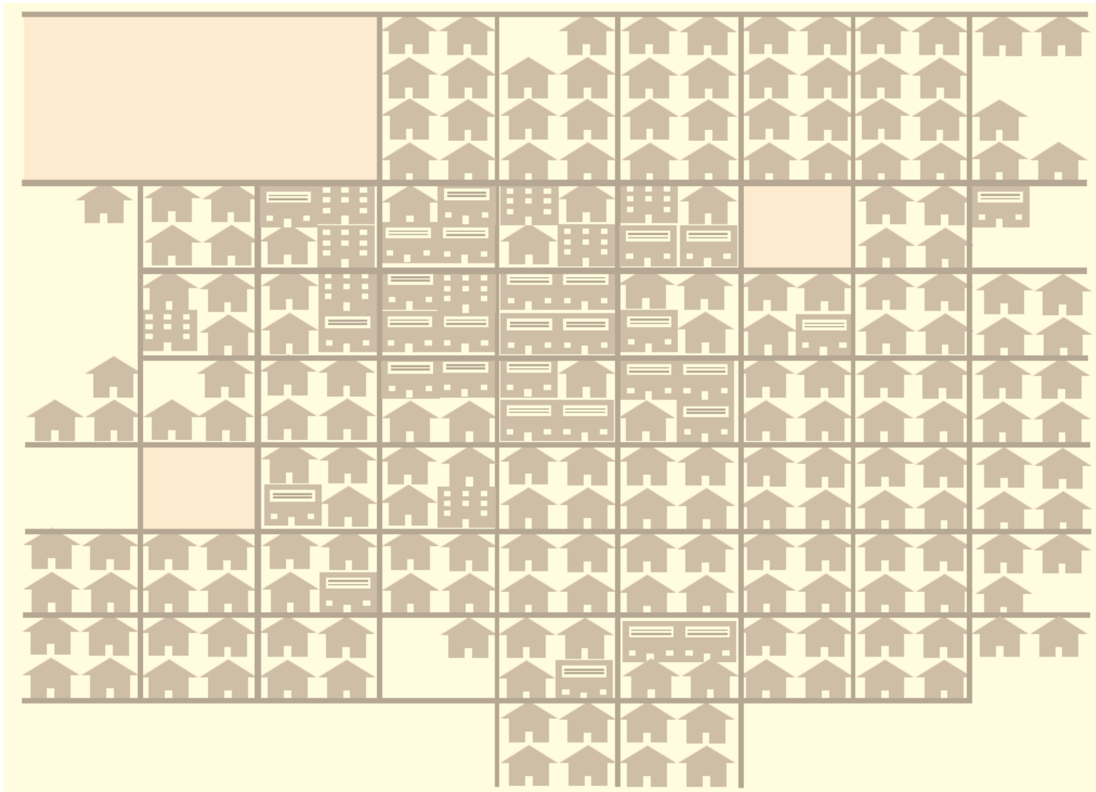


Figure 9.2: A typical *Talk of the Town* town. Buildings are represented by three basic types: houses, apartment complexes (with windows), and other businesses (with abstract signage). Empty tiles are empty lots, and the larger filled rectangles are tracts (large lots for special business use). Each town spans nine by nine blocks, and roadways are constructed as new buildings require them. Denser commercial areas emerge from a simple mechanic: prospective business owners prefer lots near other existing businesses. (Note: this image was not produced by the simulation code, but rather it was constructed manually from a generated town.)

over time by the characters who live in them. Figure 9.2 shows an example *Talk of the Town* town after several decades of existence.

Tracts, Lots, Streets, and Blocks

As I just noted, the procedure for plat generation produces a grid of variably sized rectangles, most of which are one-by-one squares—in *Talk of the Town*, these are called *lots* and the larger rectangles are *tracts*. Homes and businesses may be constructed on lots, while tracts can become the sites of specific kinds of businesses (cemeteries, parks, farms, quarries, and coal mines). As they become needed by new buildings, *streets* are constructed to follow the town's plat grid; they do not cut through tracts, as Figure 9.2 illustrates. Streets are additionally broken into *blocks*, which are short segments with two lots on either side. When a street is constructed, it is named by a probabilistic procedure by which there is a chance of giving it a numbered name (e.g., `2nd Street`) or a name randomly selected from either the surnames corpus or the corpus of American place names that was also used in *World*.[14] Lots and tracts are attributed *house numbers* according to their position on the town plat, and the concatenation of a house number and a street name produces an *address* (such as `255 2nd Street`) that may be attributed to a building or other business site constructed on that land.

Homes

A character may build a *house* on a lot, as I will discuss below. In *Talk of the Town*, a house is abstractly a building, and all houses are modeled the same. The system keeps track of data surrounding the construction of a house (architect, builders, construction firm) and its former and current residents.

Businesses and Vocations

As I will explain shortly, characters may start businesses in a town. The modeling of a business integrates both the abstract idea of a company and the concrete notion of the building or site where that company is headquartered. In *Talk of the Town*, a *business type* is defined compositionally according to author-defined values for each of the following set of features:

[14] I am now realizing that I did not boost the probability of selecting 'Minneapolis' from the corpus of place names, as I did in *World*—a missed opportunity.

- **Advent.** A year in which a business of this type becomes era-appropriate. For instance, a tattoo parlor would be anachronistic prior to 1970 or so, and so I set the advent value for that business type to `1970`. Other business types may already exist at the beginning of the simulation, such as a farm.

- **Demise.** A year in which a business of this type stops being era-appropriate. For example, blacksmith shops should disappear shortly after 1945, so I set the demise for that type at `1945`. As I explain below, businesses do not automatically shut down when a demise year is reached, but rather the probability of that happening increases dramatically.

- **Minimum population.** The minimum population that is necessary for a business of that type to operate in a town. For example, it does not make sense to open a department store in a town with few residents, so I set the value for this feature to `200` in defining that type.

- **Maximum number.** The maximum number of businesses of this type that may simultaneously operate in a given town. This value captures that a small town could potentially have several taverns or restaurants, but not two hospitals, for example.

- **Land requirements.** Whether a business of this type operates on a tract or a lot. For instance, farms requires tracts, while taverns operate on lots.

- **Services.** The services that are provided to town denizens by a business of this type, e.g., `furniture` or `dairy`. These are used to guide the simulation of character routines.

- **Owner.** The vocation of an owner (and thus founder) of a business of this type. For instance, a blacksmith shop should be owned by a blacksmith, and so I defined this value as `blacksmith` in defining that business type.

- **Day shift.** Vocations for a minimal set of employee positions that are needed for a business of this type to operate during a day timestep. In the case of a bank, I defined this as `(bank teller)`.

- **Night shift.** Vocations for a minimal set of employee positions that are needed for a business of this type to operate during a night timestep. Banks are not open at night, but they may have a janitor there to clean up; as such, for that business type I defined this feature as `(janitor)`.

- **Supplemental day shift.** The vocations of additional daytime employee positions that may be hired for in the case that an owner of a business of this type seeks to expand it. Here is how I set this feature in the definition of a bank business type: `(bank teller, bank teller, manager)`.

- **Supplemental night shift.** The vocations of additional nighttime employee positions that may be hired for in the case that an owner of a business of this type seeks to expand it. In the case of a bank in a small town, one janitor is probably enough, so I defined this feature value as an empty list.

- **Public.** Whether a business of this type is a public utility, such as a cemetery, park, or city hall (as opposed to a bar, bank, or restaurant, for instance). Public companies have no owners.

In the codebase that was used in *Bad News*, I authored the following fifty-two business types: `apartment complex`, `bakery`, `bank`, `bar`, `barbershop`, `blacksmith shop`, `brewery`, `butcher shop`, `candy store`, `carpentry company`, `cemetery`, `city hall`, `clothing store`, `coal mine`, `construction firm`, `dairy`, `day care`, `deli`, `dentist office`, `department store`, `diner`, `distillery`, `drug store`, farm, fire station, foundry, furniture store, general store, grocery store, hardware store, hospital, hotel, inn, insurance company, jewelry shop, law firm, optometry clinic, painting company, park, pharmacy, plumbing company, police station, quarry, realty firm, restaurant, school, shoemaker shop, supermarket, tailor shop, tattoo parlor, tavern, and university.

Beyond the above features, the system tracks data about businesses pertaining to its founder, current owner, former owners, current employees, former employees, founding, building construction, closure, and building demolition. Of course, businesses also have names, and these are attributed by probabilistic procedures that I defined for each business type. A generated name may include the name of the town (Duluth Dairy), a street (5th Street Diner), or the business owner or employees (Law Offices of Mateas and Wardrip-Fruin), or it may reference an emergent town element that is associated with that land. For example, a new park that is established an the site of a former farm may be named for the farmer who had tended to the land, e.g., Horswill Farm Park or simply Lessard Park.[15] In the case of a bar or restaurant, a name is determined by selecting from a corpus of evocative bar and restaurant names that I found online. The *Bad News* aesthetic, familiar to anyone who has experienced the piece, is derived in part from the peculiar monikers that adorn its generated bars and restaurants—one performance, for example, centered on a rivalry between two restaurants called Prawn Logic and Fish Matrix.

[15]This is probably my favorite system detail. As I have expressed at various times in this document, I am interested in feedback loops between emergent phenomena, and one interesting kind of feedback loop connects units of emergent content whose procedural generation is decoupled in the simulation procedure. In the case of farm naming, a little piece of procedurally generated content—a string constituting the name of a storyworld landmark—captures what has come before in the storyworld, and more specifically, it explicitly references an earlier unit of procedurally generated content (a character name). Now, in *Hennepin*, I am exploring the generation of artifacts that capture past events so that characters who encounter those artifacts come to learn about the captured events. This is the way toward *generative environmental storytelling*, as I have noted multiple times above.

Vocations themselves follow a simple representation scheme that entails the definition of values for the following features:

- **occupational status.** An integer between 1 and 5 that captures the prestige of a particular vocation. For example, `apprentice` is attributed a level of `1`, `quarryman` a `2`, `jeweler` a `3`, `doctor` a `4`, and `owner` (e.g., of a bank) a `5`. As I will explain below, other characters recognize the social status of high-prestige occupations, and occupational status also figures heavily in character decision making surrounding the hiring process. Moreover, when a character is generated outside the town to move there for occupational reasons, the character's age will be sampled from a normal distribution associated with the occupational status (this captures a general positive correlation there).

- **Employability conditions.** This is a simple function that takes a character as an argument and returns whether that character would be employable in this vocation given the current era of the storyworld. This is meant to prevent anachronisms such as a female police chief in the 1860s, for instance. I could have authored the simulation to instead be utopian, but my approach was to target a general verisimilitude so that players could bring their intuitions about rural American life (and its history) to bear in playing *Talk of the Town* (2015), and ultimately *Bad News*. As such, I was aiming for a particular kind of *SimCity effect*—Noah Wardrip-Fruin's term for cases in which "a system [...] brings the player to an accurate understanding of the system's internal operations" [1310, p. 2]—by which humans may more easily understand systems whose operations match their *preexisting* mental models about a subject phenomenon. Some have taken exception to this design decision, but I think I stand by it.[16]

- **Degree requirement.** Whether a vocation requires a college degree in order for a character to be hired—e.g., one does not become a doctor or a lawyer right out of high school. As I explain below, there is a very rudimentary modeling of characters attending college by which a probabilistic trigger may cause characters who are unemployed (and more than four years out of high school) to be awarded a degree.

In the codebase that was used for *Bad News*, I authored the following seventy-nine vocations: `apprentice`, `architect`, `baker`, `bank teller`, `barber`, `barkeeper`, `bartender`, `blacksmith`, `bottler`, `brewer`, `bricklayer`, `builder`, `bus driver`, `busser`, `butcher`, `carpenter`, `cashier`, `clothier`, `concierge`, `cook`, `cooper`, `daycare provider`, `dentist`, `dishwasher`, `distiller`, `doctor`, `dressmaker`, `druggist`, `engineer`, `farmer`, `farmhand`, `fire chief`, `firefighter`, `grocer`, `groundskeeper`, `hotel maid`, `innkeeper`, `insurance agent`, `janitor`, `jeweler`, `joiner`, `laborer`, `landlord`, `lawyer`, `manager`, `mayor`, `milkman`, `miner`, `molder`, `mortician`, `nurse`, `optometrist`, `owner`, `painter`, `pharmacist`, `plasterer`, `plastic surgeon`, `plumber`, `police chief`, `police officer`, `principal`, `professor`, `proprietor`, `puddler`, `quarryman`, `realtor`, `seamstress`, `secretary`, `shoemaker`, `stocker`, `stonecutter`, `tailor`, `tattoo artist`, `taxi driver`, `teacher`, `turner`, `waiter`, `whitewasher`, and `woodworker`.

[16] Of course, subverting preexisting mental models can be a powerful design move, especially when such mental models are rooted in prejudice or misunderstanding. To name just two examples, Mary Flanagan's book *Critical Play* articulates a poetics of what she calls 'radical game design' [343], and Vi Hart and Nick Case's 'playable' *Parable of the Polygons* [457] operationalizes game theorist Thomas Schelling's model of segregation [1110] to upturn interactor misconceptions pertaining to the phenomenon. While *Bad News* is subversive in other ways—in an early paper on the project, we wrote about how it subverts the treatment of death in videogames [1062]—it does not aim to critically interrogate American society, culture, or history, and this design (dis)inclination has worked to structure *Talk of the Town*.

Events

As I will explain shortly, the *Talk of the Town* simulation is primarily concerned with the generation of *events*, generally in the sense of character life events. In the simulation code, event definitions bundle all of the functionality that is needed in order to model the transpiring of an event and its effects on the storyworld. Critically, the system also tracks metadata for past events, including the date and location of occurrence, as well as any other event-specific information that may be useful to reason about later—in the case of *Bad News*, this is information that I may query in my capacity as 'wizard' (see next chapter).

Because it stores representations of past events so that they may be reasoned about later, *Talk of the Town*'s chronicling far exceeds that of *World*, which makes the task of story sifting—one of my jobs as *Bad News* wizard—more feasible. Indeed, it would be very hard to be a human story sifter for a *World*-driven experience. Still, *Talk of the Town* fails to do causal bookkeeping. This makes automatic story sifting a difficult prospect, since a computer program lacks my common sense about what kinds of events tend to cause others, let alone my expert design knowledge as the person who crafted the simulation. I will return to this deficiency in the course of my *Bad News* case study in the next chapter.

Artifacts

Like its predecessor *World*, *Talk of the Town* also features character gravestones, though with an added flourish: inscriptions. Specifically, a gravestone inscription is a string that is generated by the following template:

```
[header]

[name]
[birth year - death year]
```

```
[optional occupation inscription]
[optional family inscription]
[optional epitaph]
```

An example header is the canonical 'Rest in Peace', while an epitaph is something like 'Peace is thine and sweet remembrance is ours'; in each case, an instance is selected from corpora that I found online. In a bygone era, American gravestones often featured the vocation of a deceased male in the case that the occupation was central to that person's identity [482]—this practice of including *occupational inscriptions* is operationalized in this system. Likewise, the optional family inscription may capture the relation of the character to any family members that survive in the town (a practice that is extant in the real world). Here are example gravestone inscriptions for actual *Talk of the Town* characters who were buried in a mining town called Kessenich:

```
Loving memories last forever

Mable Lena Topham
1816-1904

Loving Wife and Mother
Beautiful memories left behind
```

```
                            Here lies buried

                     Charlie Joseph Gradert
                                  1830-1900
```

```
In remembrance of

Herbert Jessie Leuthold
1901-1973

Miner
Loving Husband, Father, Grandfather
Memory is a golden chain that binds us till we meet again
```

Additionally, there is one other kind of artifact in *Talk of the Town*: wedding rings, which are of course worn by characters who are married. In each case, the artifacts encode an emergent simulated event—either a death or a marriage—and in the case of a gravestone, the content instantiated in the artifact captures more emergent information (such as a character's occupation or family relations). Again, in designing a simulation I seek to target such emergent feedback loops.

Artifacts hook into the system's modeling of character knowledge phenomena, as I will explain in more detail below. When a character examines an artifact, knowledge about its subject is transmitted and the examiner's beliefs are updated accordingly. In this way, each of the components of a gravestone inscription are data-rich, in that they encode information about the storyworld that a non-player character can actually glean.[17]

Another impetus for modeling artifacts was to provide critical information and intrigue to *Bad News* players, who are tasked with learning about a character's family history. To my continuing astonishment, however, no player has ever taken the liberty of visiting a town's cemetery. As for wedding rings, these likewise provide valuable information, since knowing a character's marital status is critical to ascertaining the identify of her next of kin. However, my collaborators and I discovered through playtesting that the expression of whether characters wear wedding rings actually disrupted the balance of the experience (players learned

[17]For last year's Procedural Generation Jam (better known as PROCJAM), I gave an invited remote presentation titled "Beyond Vending Machines: Character–Content Feedback Loops" (November 8, 2017). In the talk, I identified a design pattern whereby procedurally generated content actually changes the constitution of simulated characters in meaningful ways. By targeting such *character–content feedback loops*, a system may model the internal lives of characters who have meaningful ongoing experiences in their simulated storyworlds; moreover, such feedback loops between generated content and future generated content can propel storyworld instances into novel pockets of a simulation engine's possibility space. While I am now more deeply exploring this notion in *Hennepin*, the implementation here of procedurally generated data-rich gravestone inscriptions—ones that actually *change* characters who examine them—represents an initial attempt at following this design pattern.

too much too early) and so I removed that component of character descriptions. As such, characters do wear wedding rings in the simulation, but this is never expressed at the surface of *Bad News* gameplay. Now you know.

I also planned to implement other artifacts such as maps, documents, photographs, and paintings, all of which would encode knowledge about the world that characters could glean by examining the artifacts. As I will explain in Section 11.2.11, such extensive artifact modeling is now implemented in my new system *Hennepin*, where artifacts may encode information about past events.

9.2 Simulation

Now that I have explained the basics of *Talk of the Town*'s entity modeling, in this section I will outline its simulation procedures. At a high level, the workflow here is the same as that of its predecessor *World* and its successor *Hennepin*: create an empty world, have some initial characters move into it, and then proceed from there, timestep by timestep, simulating storyworld phenomena by executing a recurring simulation loop. As I have noted above, by this configuration, there are two distinct modes of simulation, one pertaining to the initial setup of a storyworld and the other to its evolution over time—together, these constitute the simulation engine's procedure for *world generation* [751]. I will describe both of these simulation modes in turn.

9.2.1 Setting Up a Town

Each *Talk of the Town* storyworld begins with the same essential scenario: it is the summer of 1839 and a few families move into an empty American townscape to establish a farming community; just prior to its establishment, a plat is generated for the town following the procedure explained in Section 9.1.3.

Retconning

As with *World*, modeling this initial scenario requires some retroactive simulation, in this case pertaining to the lives of the initial town denizens prior to their moving into the storyworld. Essentially, this entails determining birthdates for these *primordial* characters and simulating a history of childbirth in the case of a married couple, which produces children who will also be among the town's founding citizens. The number of families that move into the town will depend on the number of tracts there—one tract will be reserved for a town cemetery, and the remainder will become family farms.[18] Additionally, there is a small chance of one of the tracts being used for a coal mine or a quarry instead of a farm, in which case the place will take a form more akin to a company town than a farming community. Beyond the characters who will own the farms or companies (and their family members), other characters will also move to the town to work there as farmhands or cemetery workers, or in other capacities in the case of a coal mine or quarry being founded at the outset of the town's history.

A Simple Block of Code

As I will explain more thoroughly in Section 9.2.7, the entire procedure of creating an initial set of characters to live in the town is driven in a elegant way by the simple instantiation of a new business. For example, consider the following block of Python code:

```
# Create character
farmer = PersonExNihilo(
    sim=self, job_opportunity='farmer'
)
```

[18]If a generated plat contains less than two tracts, a new one is created to replace it.

When this code is executed, a new character is created outside the town—the class `PersonExNihilo` (lit. 'person out of nothing') represents characters who are not born from parents in the simulation—to become a farmer there. That is, they will be created according to a recipe for what makes an American farmer of 1839; this recipe is composed modularly according to the definition of the `farmer` vocation, which specifies a typical age, sex, and family status for various eras. By executing this code, the execution of other code will be triggered, which may cause additional family members to be created in turn. Once all of the characters' backstories have been simulated, they will seek residence in the town—because no homes will exist yet, they will have to select an empty lot and construct a house (according to the procedure that I will outline in Section 9.2.8). Having constructed a house, they will move into it, and thereby the first characters will come to live in the town.

Here is the next line of code that will be executed in the procedure that is enacted to set up the town:

```
# Have that character start a farm in town
Farm(owner=farmer)
```

When this line is executed, other code will be triggered to handle the character selecting an empty tract on which the farm will operate (likely the one closest to the character's new home), and critically the hiring of an initial farmhand. By the latter procedure, the character will attempt to first hire someone in the town, but no one in the town will be qualified, so the simulation will generate a new character from outside the town to serve as a farmhand. This will trigger an enactment of the same process whereby the farmer came to exist and eventually live in the town. This process is repeated for all of the initial farms and businesses that will exist in the town, and once everyone has moved in accordingly, the town's history is set in motion.

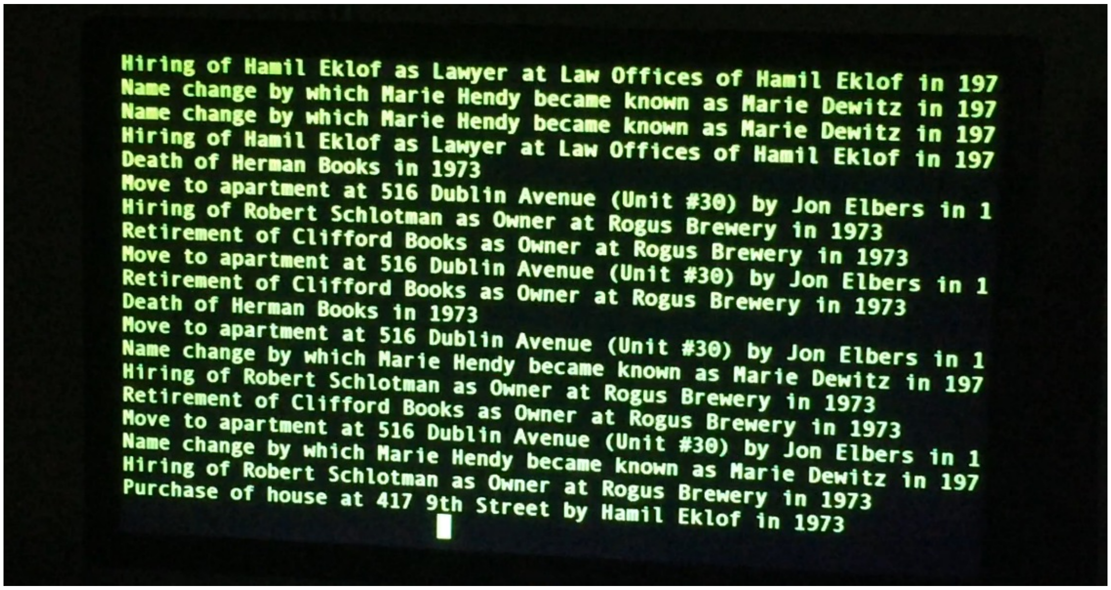


Figure 9.3: Console output from a *Talk of the Town* simulation instance. In motion, the output is a dizzying flurry of events sampled from the history of the town as it is transpiring in real time. While it is not readable, this display is utilized during *Bad News* performances to provide visual intrigue to audiences who may congregate around the *wizard command center* of an installation.

Town Seeds

Eventually, I set up a mechanism for *seeding* towns—i.e., initializing the simulation's random number generator by using a numerical *seed* [744]—which allows for the same town to be generated again by using the same storyworld seed. This was, however, not implemented in the version of the simulation codebase that is used for *Bad News* (which means that *Bad News* storyworlds are irretrievable).

9.2.2 Simulation Loop

Once a town has been established, its history is simulated until a specified *termination date*; in the case of *Bad News*, this is August 19, 1979.[19] As with *World*,

[19] I was keen on setting *Talk of the Town* (2015) specifically in the summer of 1979, due to it being an aesthetically pleasing number, and moreover because it solved a design issue that

time progresses in *Talk of the Town* according to the execution of simulation loop that recurs each timestep.

Level-of-Detail Modulation

As I noted above, not every timestep is actually simulated—more precisely, not every timestep is simulated *at full granularity*. For one, character knowledge phenomena is not simulated until after world generation ends—i.e., after the termination date has been reached—because it is too computationally expensive (both in terms of memory usage and speed). What may or may not be enacted on a given timestep is the procedure that drives social simulation; the probability of carrying out social simulation on a timestep is parameterizable, and in the case of *Bad News* it is set such that approximately four days will be simulated a year. Regardless of whether social simulation will be enacted on a timestep, various other high-level phenomena may be triggered. Figure 9.3 shows the raw console output that is displayed on screen as this procedure is underway. In the following sections, I will explain all the various phenomena that are simulated over the course of a *Talk of the Town* simulation loop. While these phenomena are grouped into more specific subroutines in the actual code for the simulation loop, for clarity I will describe them in terms of larger conceptual groupings.

9.2.3 Birth and Death

Just as in *World*, a character's time in the simulated storyworld is bookended by *birth* and *death* events, whose modeling I will explain in this subsection.

Adam Summerville and I identified: gameplay could break down were it set in the current day, because cell phones and the internet would yield hyperactive knowledge propagation. Since this was still a concern in the context of the *Bad News* design, the summer of 1979 stuck.

Birth

When a character becomes pregnant, the date that she will give birth is established at that time. This is kind of like a 'due date', except that it represents the actual date of birth (using the simulation cheat of *foreordaining*). Once that date arrives, there is a 50% chance that the birth will occur during the day timestep, otherwise it will happen at night. The birth event itself is abstractly simulated by the parents soliciting a doctor to delivery the baby (if there are any in town) and then giving birth at the hospital (if there is one in town), where any nurses who are on duty will also become associated with the event. Once the baby is born, the parents name her by the procedure outlined in Section 9.1.2, which includes decision-making about whether to give the infant a hyphenated surname. Next, the characters take the baby home, which triggers updates to that character's home attributes (including those that track residential history). Finally, depending on the era and whether there is a daycare center in town, the mother may decide to quit her job to become a stay-at-home parent.[20]

Death

Death strikes according to a probabilistic trigger, just as in *World*. Whereas a number of causes of death were defined for the latter, in the version of *Talk of the Town* that is used in *Bad News*, there is only 'natural causes'. Once a character reaches the age of 68, there is a set chance on each simulated timestep that she will die of natural causes at that time.[21] When it has been determined

[20] Of course, in the real world, fathers can also be stay-at-home parents. This was probably not typical in American small-town life prior to 1979, however, so this phenomenon is not modeled in the simulation.

[21] The oldest character that I have encountered was 111 years old, and this character emerged in a fitting context: Tarn Adams was playing *Bad News* at the San Francisco Museum of Modern Art, and the next of kin to whom he was tasked with delivering a death notification was an 111-year-old woman. This is fitting because Tarn Adams, as cocreator of *Dwarf Fortress*, is likely the person on this planet who is most associated with emergent intrigue and bizarre procedural

that a character will pass away on a timestep, an abstract simulation of the event transpires in the same style as the birth event that I have just described. If there is a mortician in the town, the next of kin contracts that character to handle the interment. This involves the selection of a cemetery plot and the placement of the deceased in that place (abstractly modeled as updating her home attributes).

If the character is married, her widow(er)'s marriage attributes will be updated and that character will begin to grieve (which alters decision making) and she will take her wedding ring off her finger.[22] If the deceased was a single parent, her adolescent children will depart the town (and thus the simulation) as an abstract modeling of the notion of being taken in by family in another place. If the deceased owned a home and a grown child lived with them at the time of death, ownership of her home will be transferred to the child. Finally, if the deceased character was still working at the time of death, her company will hire someone to fill her vacated position; this may trigger the generation of a new character to move into the town, per the procedure outlined in Section 9.2.1.

9.2.4 Character Routines

The primary driver of character simulation in *Talk of the Town* is a procedure that handles *character routines*. On each timestep that will be simulated at high-granularity, every character in the town decides where to go; characters do not change locations within a timestep, so this decision determines where they will spend the entire day (as in daytime period) or night.

outliers. I will mention more about this story in the next chapter.

[22] I understand that a grieving widow(er) is probably not likely to do this. One of many `todo` items in the code concerns the development of a more nuanced modeling of this process, but I never got to it. As I have noted above, this chapter is about describing the system (with full disclosure) with regard to how it is actually implemented in driving *Bad News*.

Mechanic: Deciding Where to Go

To decide where to go, a character carries out a probabilistic procedure that works as follows. If the character is a child and it is daytime, she will attempt to go to either school or daycare, depending on her age; if she is too young for school and there is no daycare in town, she will stay home, and likewise if there is no school in town (because it is very early in the town's history). During nighttime, children stay at home. If the character is an adult and the time of day corresponds to her work shift, she will go into work (unless she will call in sick, which happens at a low probability).[23]

If an adult character does not have to go to work, then she may decide to stay home, the chance of which depends on her personality (extroversion and openness) and whether there are kids at home to look after. If a character decides to leave home, she then probabilistically determines whether she will visit another character or visit a business. In the former case, another procedure determines whom she will visit, based on concerns such as whether the candidate is a neighbor, a family member (and if so, what relation), a friend, or a coworker, but namely her affinity for the character.

To decide which business to visit, the character considers all the services that are provided by each (see Section 9.1.3) and selects the one she desires probabilistically, depending on the time of day—that is, for both daytime and nighttime I authored a set of relative utilities for each type of service, to which a probability distribution can then be fit in order to select one probabilistically. These relative utilities capture that, for example, during daytime one is more likely to seek baked goods than a bar, but at nighttime that relation is inverted. Once a service has been selected, the character compiles all the businesses in town that

[23]Weekends are not modeled in the simulation, which means that every day works according to the same simulation procedures.

provide that service and then selects which one to visit probabilistically according to the distance from her home. In a sense, town locales in this procedure are like the *smart objects* [348] of *The Sims* [792], which are storyworld objects (including characters) that *advertise* affordances for character interactions—for instance, a refrigerator advertises eating (more precisely, lowering hunger) while a nearby character advertises social interaction (which satisfies a social need).

Dynamic: Emergence of Neighborhoods

From this procedure, distinct neighborhoods emerge in a way that is centered on public establishments (what sociologist Ray Oldenburg calls *third places* [896, 897]): characters tend to visit the establishments in their proximity, and so the characters in a given neighborhood will tend to congregate at the hot spots that are located there; in turn, according to the system's social simulation (discussed in the next section), characters will build relationships with the people that they encounter at those hot spots. This feedback loop is also driven by the social mechanic of characters being more likely to visit neighbors and friends. Generally, *Talk of the Town* is about utilizing simple social mechanics to yield rich social dynamics, as the rest of this chapter will emphasize.

9.2.5 Social Simulation

Social simulation in *Talk of the Town* works according to a basic principle: characters' (nonreciprocal) *affinities* for one another evolve as a function of the compatibility of their personalities and the amount of time that they spend together.[24] In this way, friendships emerge as compatible characters interact more extensively and enmities emerge as incompatible characters do so. This aspect of

[24]The subsystem reported in this section was first reported in an earlier workshop paper [1043], though the prose here is significantly revised.

the subsystem extends *World*'s approach to social simulation, which I described in Section 7.3.4. Beyond platonic affinity, character romantic feelings evolve by the same principle (applied to romantic considerations).

Evolving Social Networks

Over time, this abstract social simulation evolves a *social network* in the town that interconnects its denizens according to their social and romantic feelings for one another, as well as the distinct relationship types of `aquaintance`, `friend`, and `enmity`. This network, along with the intermeshed family and work networks of a town, form the social substrate that character knowledge networks ride atop, as Section 9.2.12 illustrates.

Character Affinities

Social simulation in *Talk of the Town* is driven by several concerns (about which more soon), but the two core notions are called *charge* and *spark*:

- **Charge.** A nonreciprocal *platonic affinity* held by one character toward another, represented by a continuous numerical value. This value may be positive, signaling feelings of friendliness, or negative, indicating feelings of enmity. Broadly, charge increases as a character spends more time around someone who is compatible with her and decreases as she spends more time around a character who is incompatible with her. When little or no time is spent with someone, charge remains neutral.

- **Spark.** A nonreciprocal *romantic affinity* held by one character toward another, represented by a continuous numerical value. This value may be positive, signaling romantic attraction, or negative, indicating romantic aversion; additionally, a default value (`n/a`) represents the absence of such feelings, in cases where romantic attraction is not applicable. As I will explain shortly, spark evolves similarly to charge.

Additionally, it is critical to note an additional notion, *salience*:

- **Salience.** For each character in the town, the system tracks the salience of all other characters to that person. Salience values are on a continuous scale, and the procedure for calculating them captures that for example, one's best friend is more salient than an acquaintance, a neighbor who is also a coworker is more salient than another neighbor, and so forth. Additionally, characters with higher occupational statuses are more salient to others, and salience increases slightly for each social interaction one has with another character.

When two characters interact on a given timestep, their respective affinities for one another will evolve according to how *compatible* each is with the other:

- **Charge increment.** As characters spend more time together, their respective charge values for one another will increase or decrease according to a notion of *friendship compatibility*, which is nonreciprocal. Specifically, this notion is used to determine a *charge increment*, which specifies how much a character's charge for another will increase (or decrease) in the case of a social interaction. In *Talk of the Town*, charge increments are modeled by operationalizing the following relationships between the five-factor personality model and friendship formation that have been identified by Maarten Selfhout and collaborators [1126] and Lois Verbrugge [1293]:

    1. People with similar openness, extroversion, and agreeableness personality components are more likely to befriend one another [1126].

    2. People with higher extroversion select friends more frequently [1126].

3. People with higher agreeableness are selected as friends more frequently [1126].

4. People of the same sex are more likely to become friends [1293].

5. People closer in age are more likely to become friends [1293].

6. People of the same occupational status are more likely to become friends [1293].

To calculate the two respective charge increments for a pair of characters, the system carries out the following procedure (numbers here indicate which finding from above each step operationalizes):

1. Calculate a base charge increment (for both characters) as the average difference between the characters' openness, extroversion, and agreeableness personality components, and then normalize this average to a scale between `-1.0` (maximum disagreement) and `1.0` (maximum agreement).

2. Modulate the charge increment held by each character according to the character's own extroversion personality component: if she is extroverted, boost the charge increment; if she is introverted, reduce the increment.

3. Modulate the charge increment held by each character according to the *other* character's agreeableness personality component: if the other character is agreeable, boost the charge increment; if she is disagreeable, reduce the increment.

4. Reduce both charge increments if there is a sex difference between the characters.

5. Reduce both charge increments commensurately to the difference in age between the characters.

6. Reduce both charge increments commensurately to the difference in occupational status between the characters.

- **Spark increment.** Like with charge, as characters spend more time together, their respective spark values for one another will increase or decrease according to a nonreciprocal notion of *romantic attraction*. Specifically, this notion is used to determine a *spark increment*, which specifies how much a character's spark for another will increase or decrease in the case of a social interaction. Again, the system operationalizes findings from the social sciences. In this case, these are the following relationships between the five-factor personality model and romantic attraction that have been identified by Shanhong Luo and Guangjian Zhang [717], Andrew Hayes [460], and John Marshall Townsend and Gary D. Levy [1263]:

  1. People with higher openness, extroversion, agreeableness, and neuroticism are more likely to form romantic attractions [717].

  2. People with higher conscientiousness are less likely to form romantic attractions [717].

  3. People with higher conscientiousness, extroversion, and agreeableness are more attractive to others [717].

  4. Romantic attraction decreases with age difference [460].

  5. People with higher occupation status are more attractive to others [1263].

  Interestingly, Luo and Zhang found no significant correlation between attraction and personality similarity [717], and so such correlation is not operational in *Talk of the Town*. To calculate a spark increment for one character relative to another, the system carries out the following procedure (numbers again indicate which finding from above each step operationalizes):

0. If romantic attraction does not apply—because the character is not attracted to that sex, or they are family members, or attraction is inappropriate given the age of one or both—then set the spark increment to `n/a`; more precisely, in the Python code, this is `None`. As I will note below, this step introduced unintended behavior that would produce a hallmark of the *Bad News* aesthetic.

1. Calculate a base spark increment according to the character's own openness, extroversion, agreeableness, and neuroticism personality components. Here, the system follows the sex differences of Luo and Zhang, and in fact uses the very numbers that they report in their paper [717, p. 948]. This is a good example of the appeal of using the five-factor model: sometimes a paper will essentially include pseudocode for modeling a human phenomenon that is driven by personality factors.

2. Modulate the spark increment according to the character's own conscientiousness personality component: if she is conscientious, decrease the charge increment; if she is not, boost the increment.

3. Modulate the spark increment according to the *other* character's conscientiousness, extroversion, and agreeableness personality components: for each, higher values yield commensurate boosts and lower values produce corresponding decreases.

4. Reduce the spark increment commensurately to the difference in age between the characters. While Hayes reports sex differences, this nuance is not operationalized in the system.

5. Boost the spark increment commensurately to the other character's occupational status. While Townsend and Levy report sex differences, this nuance is not operationalized in the system.

Intergenerational Romance

The first time a pair of characters meet, the system will calculate the respective charge and spark increments by following the procedures that I have just outlined. One unintended (and intriguing) system behavior arises from the procedure for determining a spark increment, as I have alluded to above. When two characters meet prior to reaching the age in which romantic attraction begins, the system will attribute a spark increment of `n/a` in each direction. This makes sense and produces the intended behavior initially, but a problem arises in that these increments do not change over time, a ramification of which is that characters who meet early in life (e.g., in school) will *never* be able to develop romantic feelings for one another (even in adulthood).

The perhaps unintuitive upshot of this bug is romantic relationships with massive age differences, which obtain because characters can only become romantically attracted to others whom they meet *once they have reached the age of romantic life*. Usually a first meeting like this only occurs in the case of a generational gap—since otherwise the characters will have grown up together and congregated in the same social circles (at school and in other public places)—and so age differences dominate the romantic world of *Talk of the Town*.

When my collaborators and I began performing *Bad News*, we encountered this phenomenon as a curious recurring motif: characters were constantly falling in love with their parents' friends. Initially I assumed that age difference was positively contributing in the determination of spark increments, rather than negatively contributing, but as I have just explained the strange behavior was in fact dynamical rather than mechanical. In any event, we found that the emergent quirk actually contributed dramatic intrigue to the world—players found the age gaps to be curious and worth inquiring about—and so I elected not to fix the bug in the version of the code that is used in that project.[25] These are the *aesthetics of the uncanny* at work, and this is an example of the broader phenomenon of a bug producing surface-level aesthetic quirks that become adopted as canonical.

In a similar case—though one without an intriguing byproduct—a bug was yielding strange population dynamics in the generated storyworlds: each town would gradually dwindle in its population, but never all the way to zero. After hours of fruitless investigation as to the cause of this behavior, I returned to working on the subsystem that handles character routines, where I noticed a typo in the code that made characters unable to go out for errands or for leisure. When I fixed this, I was shocked to discover that the population dynamics had reverted back to normal. At first, I could not comprehend how the two concerns were connected, but eventually it dawned on me: if characters only left their homes to go to work, their only potential romantic partners would be coworkers—but most people worked at family businesses, and so this bug had essentially disabled the system's romance simulation. As a result, few characters had romantic lives, which ultimately meant few children were ever born, and the towns' dwindling populations could not be replenished.

Simulation Bugs Are Narratable

One enjoyable aspect of crafting character simulations is that otherwise benign bugs in the code generate *narratable* surface expressions, since the bugs alter the

[25] Later, in another version of the simulation code, I fixed the romance bug by simply altering the procedure to recalculate spark increments once each story year.

behavior of anthropomorphized code objects (characters). Indeed, to use Michael Mateas's phrasing, the actual simulation execution concerns "a purely relational (and thus literally meaningless) technical manipulation of computational material" [766, p. 61]. Here, Mateas is writing about how AI approaches, all of which are instantiated in the otherwise meaningless manipulation of computational material, are actually distinguished by the coupled interpretative strategies employed by their human practitioners:

> The rhetorical strategies used to narrate the operation of an AI system varies depending on the technical approach, precisely because these interpretative strategies are inextricably part of the approach. Every system is doubled, consisting of both a computational and rhetorical machine [766, p. 61]

In the case of character simulation, the rhetorical machine drives the narration of code objects as anthropomorphic beings whose behaviors we charge with all the meaning that structures human life. Really, though, there is nothing there—the town is just an electric wind. But now I have digressed back onto my tangent on representation, rhetoric, and symbol grounding that appeared in Section 4.2.1.[26]

Procedure

The actual procedure that evolves character affinities over time works by carrying out the following steps:

- **Decay charge and spark.** First, at the start of a simulated timestep, the system *decays* all the charge values held by the characters who currently live in the town; this is done by regressing the values (whether positive or negative) toward `0`. This makes strong feelings of friendship or enmity weaken

[26]For more on representation and expressive simulation, see an underappreciated paper by Mike Treanor and Michael Mateas titled "Understanding Representation in Playable Simulations" [1266].

as time passes without the character interacting with the subject of such feelings. Without this step, childhood friendships would never fade, even if one of the pair were to leave the town, for example. In the case of spark, decay works differently: instead of reducing spark itself, the spark *increment* is decreased as two characters spend more time together. This supports romantic attraction that evolves quickly but eventually plateaus, as well as romantic obsessions that persist even without extended interaction with the target of such affection ("distance makes the heart grow fonder"). Both of these are are media tropes that I wanted the system to operationalize.

- **Trigger social interactions.** Once character routines have been enacted on a timestep, each character in the town will have navigated to the location where she will spend that day or night. In this place, she may engage in *social interactions* with nearby characters. In *Talk of the Town*, a social interaction is instigated by a probabilistic trigger that considers: a character's extroversion (extroverts instigate more social interactions), the salience of other nearby characters, and one's respective charge and spark values for those individuals (characters interact more with people they like). By considering charge and spark, the procedure yields feedback loops that may reinforce emerging friendships (or romantic crushes): characters are more likely to interact with better friends, and interactions between friends will cause friendships to be strengthened; in turn, such strengthening will increase the likelihood of future interactions between the friends, which will work to further strengthen the friendships, and so forth. This is how *best friend* relationships emerge in *Talk of the Town*.

- **Update charge and spark.** Once a social interaction has been instigated by a character, the charge and spark values that each holds for the other are updated. As I explained above, each character holds a charge and spark increment for every other character she has ever met—updating means incrementing the corresponding charge or spark by that amount, but in a way that *extrapolates* according to the amount of time that has passed since the last simulated timestep. As I noted earlier, *Talk of the Town* does not simulate every timestep in the history of a town, but rather four or so a year. By extrapolating individual social interactions, they may come to stand for extended periods of social interaction that would have transpired over the course of the interim period. To extrapolate a social interaction, the charge and spark increments are multiplied by the number of timesteps that have transpired since the last simulated one.

- **Reconstruct discrete relationships.** After the charge and spark values that one character holds for another have been updated, the system checks for whether any *affinity thresholds* have been eclipsed. For example, when a character's charge for another exceeds a certain number, the character will come to view that person as a friend, and thereby a *friendship* relationship is instantiated by the system. Because charge is nonreciprocal, this works unidirectionally, and indeed the other character may not feel the same way. In *Talk of the Town*, there are three platonic relationship types: `acquaintance` (default type, associated with neutral charge), `friend` (associated with significant positive charge), and `enmity` (associated with significant negative charge). Additionally, three superlative relationship types are tracked by the system: `best friend` (character for whom one holds the highest charge; must be above a positive threshold), `worst enemy` (character for whom one holds the lowest charge; must be below a negative threshold), and `love interest` (character for whom one holds the highest spark). Because of decay, it is also possible for relationships to be demoted over time—for example, a childhood enemy who leaves the town may be considered a mere (former) aquaintance after the passing of decades.

By this procedure, character feelings for others evolve over time, enabling the emergence of social networks that capture platonic affinities, romantic attractions, and discrete social relationships. At any point in the history of a generated town, each character will harbor a specific degree of both platonic affinity and romantic attraction for every other character in the town (though both values will be neutral in the case that the characters have not met). Below, in Section 9.3, I will discuss some interesting emergent phenomena that are produced by *Talk of the Town*'s mechanics for social simulation.

Related Work

A particularly related approach to social simulation has been reported by Gonzalo Méndez, and Pablo Gervás and Carlos León [830, 831]. This project also targets a lightweight design and features a low-fidelity simulation procedure that evolves mutual character affinities over the course of many social interactions.

In their system, character affinities evolve according to abstract social exchanges in which one character proposes an interaction to another character, who then chooses whether to accept or reject the proposal. The potential interactions that characters may engage in depend on their mutual affinities (for instance, only dating characters may have a romantic dinner together) and the various interactions affect affinities differently (a romantic dinner boosts affinities more than watching TV together does, for example). If the recipient accepts an interaction, both of the characters' mutual affinities increase; upon rejection, the proffering character will lower her affinity toward the recipient. As such, it works like a lower-fidelity version of *Comme il Faut* (*CiF*) [804], the engine underpinning *Prom Week* [799, 803], which I discussed at several points in Part I of this thesis. While social exchanges in *CiF* have a rich array of preconditions and postconditions and are rendered in natural language, exchanges in the system under discussion have simple conditions and are only represented symbolically. In sacrificing fidelity, their system gains in computational efficiency, which affords the generation of large social networks, a task for which *CiF* could prove too heavyweight.

Talk of the Town's social simulation takes this lightweight approach further, with exchanges that are even more abstract: interactions in the system are represented merely as functions that evolve affinities. While Méndez, Gervás, and León's system has been applied to social networks comprising fifteen characters [831], storyworlds in *Talk of the Town* generally contain on the order of several hundred characters. Another distinction is that their system models a single notion of affinity, subsuming both platonic and romantic feelings, while these are decoupled in my project. While *Talk of the Town*'s lightweight social simulation proved to be a good fit for *Bad News*, which requires storyworlds to be generated quickly as players arrive for their scheduled performances, I am now jumping across the continuum to model fine-grained character actions in *Hennepin*. As I explain in Chapter 11, I still produce storyworlds with hundreds of characters, which means this increase in granularity comes at steep computational costs. Accordingly, *Hennepin* can probably only drive second-order media experiences for which it is alright to take a day or so to generate a storyworld. *Sheldon County*, the subject of Chapter **??**, has been designed to work under this constraint.

9.2.6 Marriage and Divorce

Unfortunately, *Talk of the Town* does not model the romantic lives of characters beyond the triggering of marriage and divorce events. Partly this is due to the abstract modeling of social interaction in the system, but it still would have been possible to introduce a discrete dating relationship into the simulation—this was high on my `todo` list, but I never got to it. As I explain in Chapter 11, *Hennepin*'s major departure from *Talk of the Town* is its modeling of fine-grained social interactions. One outcome of this increase in detail is the actual simulation of romantic life, which has been long overdue in my simulation practice—in this case of this simulation engine, I consider it a modeling gap (see Section 4.1.5).

In *Talk of the Town*, a social interaction between two characters may trigger either a marriage event or a divorce event. The mechanism that handles this is embarrassingly simple given the importance of such life events, especially in the domain of an American small town. To that point, a comment above the code block that handles this behavior states:

```
# TODO fix currently dumb way of 'proposing'
# and divorcing
```

Marriage

Here is how marriage triggering works: if both of the respective spark values among two characters exceed an authored threshold after a social interaction, then a marriage event is triggered.[27] Same-sex marriage is not possible, since it was not legal in the United States in the simulated period (i.e., 1839–1979). As such, and due also to the lack of simulation of romantic life aside from marriage, in *Bad*

[27] I had assumed this at least worked by a probabilistic trigger, but I see now that it does not. This appears to be a case of placeholder code becoming calcified in a framework.

News the scenario of a queer character who remains in the town is tragical (as Section 9.3 will show). When a marriage event is triggered, a probabilistic procedure determines whether a newlywed or stepchildren may take a new surname, including a hyphenated construction. Additionally, the newlyweds determine new living arrangements: if either owns a home, they move there, otherwise they obtain a home in the town according to the procedure outlined in Section 9.2.8.

Divorce

If a married character falls deeply in love with someone else, a divorce will be triggered. Specifically, if a married character's spark for another character exceeds twice that of her spark for her spouse, then divorce occurs. This does not mean that the character will automatically marry her love interest—that character may not even feel the same way—but it of course opens up the possibility. Again, I view this modeling as embarrassingly abstract, and an intellectual (and technical) thrust of my development of *Hennepin* has been to actually do romance simulation.

9.2.7 Business Operations

Business operations constitute the primary structural scaffold for simulated life in a *Talk of the Town* town. With each new timestep, there is a chance that a new business may open in the town or that an existing business may shut down; here, the respective probabilities are authored simulation parameters.

Business Establishment

If the opening of a new business is triggered, the system randomly selects a business type and then checks whether it could successfully operate given the current state of the town. Here, success is contingent on: the `advent` year having been reached, the `demise` year having not been reached, the town exceeding the business type's specified `minimum population`, and there not already being enough businesses of this type in the town (as specified by the type's `maximum number` attribute).[28] Additionally, if the business will require a tract to operate, there must be an open tract in the town. If the randomly selected type is not applicable in the town, another type is selected and evaluated in the same way; this process terminates after a set number of failures.

If the system does successfully select a type of business to open in the town, the next matter is deciding who will own the business. In the case of a public business type, this is not necessary, but for all other business types the system will reason over the characters in the town to determine if anyone is a good fit for the vocation of the owner position. Here, the system operates over concerns such as the presence of a college degree (if the owner vocation requires a degree), occupational status, and whether a character is employable in that vocation given the current historical era. If no one in the town is qualified, then an adult character is generated to move into the town to start the business.

Once a character has been selected to run the new business, she must name the company, obtain a lot or tract, contract the construction of a building there, and hire an initial set of employees. Naming follows the procedure that I outlined in Section 9.1.3. To decide where to locate the business, the owner carries out a procedure of *utility-based action selection* [723, 742] whereby every lot (or tract) is scored according to a single concern: how near the location is to every other business in town. This simple mechanic causes downtown areas to emerge over time, and also secondary neighborhoods, since new businesses may be forced to locate outside a downtown area if its perimeter is taken up by residential areas (i.e., if characters build homes on lots that surround the downtown area, as in

[28]These attributes were defined above in Section 9.1.3.

Figure 9.2). Rather than selecting the top-scoring lot, the owner probabilistically selects from the top three candidates.

To actually obtain the lot, the owner *contracts* a realtor in the town by another utility-based procedure: she scores all the realtors in town according to concerns such as familial relation, affinity, and whether she has contracted that person before. (As I will note below, this procedure is used whenever a character needs to contract someone of a certain vocation.) If there are no realtors in the town, no character is contracted, but the job is still carried out.[29] Once a lot or tract has been obtained, the new business owner carries out the same procedure to contract an architect (who works at a construction firm) to construct a building on the lot. This will trigger a construction event whereby a new building is constructed; the system tracks data about this event such as its date of completion, the character who contracted the work, the architect, and the builders who carried out the work.

Now that the character has constructed a building on her acquired lot, she must hire an initial set of employees to work at the business. To do this, she considers each of the vocations that are listed in the business type's `day shift` and `night shift` attribute definitions. For each of these vocations, the owner assembles a list of candidates given all the other characters in the town. These are characters that meet the criteria for taking on that vocation—which I listed above in explaining how the system selects a new business owner—but overqualified candidates (with occupational statuses equal or greater to the one at hand) are not considered.

Once the job candidates have been assembled, the owner scores each according to whether: she already works at the company, she is in the owner's immediate family, she is in the owner's extended family, she is a friend of the owner, she is an

[29]This could be taken as implying that the character purchases the lot without the help of a realtor, but really the realtor does not actually do anything anyway. There is no concrete economical simulation carried out by the system, and the modeling of contracts is meant as a cheap source of social intrigue.

enemy of the owner, she is an acquaintance of the owner, and her current occupational status. This simple scoring procedure yields the emergent phenomena of in-house promotion (due to preference to hire from within the company), family businesses (due to preference to hire family), nepotism (due to preference to hire friends and family), and leaving a company for a better position elsewhere (since all qualified candidates in the town are considered). Finally, the owner probabilistically selects one of her top three candidates, and that character is thereby hired for the position. Because overqualified candidates are excluded and having the same occupational status as the position is treated as an overqualification, the opening represents a promotion in status for every character in the candidate pool. As such, though the procedure only explicitly models the decision making of the business owner, it may be viewed as abstractly modeling the entire hiring process whereby job candidates apply for a position, the owner interviews them to evaluate them, one is selected, and the person accepts the position.[30]

If a hired character is currently working in a different position, that position is vacated and the same hiring process is then carried out by the owner of that business (to hire a replacement); in this way, the process of hiring for one position may trigger a chain reaction in the town. Additionally, if there are no qualified candidates in the town for a given position, an adult character will be generated to move into the town. As such, even if a character in the town is hired for a position, the hiring may trigger a chain reaction that ultimately brings new people into the storyworld. In this way, the opening of a new business works as the primary driver of town evolution in terms of the actual composition of residents who live there. As such, the specification of the probability of a new business opening on some timestep is a powerful authorial lever for controlling the general population

[30]In *Hennepin*, which features fine-grained character actions, each of these steps are concretely realized, as I explain in Section 11.2.1.

dynamics of the engine's generated worlds.

Business Operation

Once all of the initial employees have been hired for a new business, it commences operation. This means that all of the employees will start going to work there on their specified shifts, following the simulation of character routines that I discussed in Section 9.2.4. Additionally, the business will now provide the services associated with its type, which means characters may start visiting it as their routines bring them out into public.

There are two attributes in the definition of a business type that I have not yet discussed: `supplemental day shift` and `supplemental night shift`. As I explained in Section 9.1.3, these are lists of additional employee positions that are not necessary for the business to operate. These are filled over time as the owner of a business does favors for friends and family members who are out of work. Specifically, when a character who is old enough to work (and not retired) is unemployed for an extended period, a procedure determines whether any business owner in town would be willing to hire them as a favor. In this way, for example, the children of a business owner may be given low-level positions at the parent's company once they reach a working age. Over time, however, due to how owner hiring preferences favor family members and internal promotion, the children may rise the ranks in the company and eventually take it over when the owner retires or passes away. This is the mechanism by which family businesses may be sustained over generations, which is a distinctive feature of American small towns and, in turn, the *Bad News* aesthetic.

Finally, characters may not be fired and only quit their jobs as part of the process of being hired for a better one; however, characters may retire according to a probabilistic trigger, which kicks in once the character reaches the age of 65. If the owner of a business decides to retire, she first hires her replacement.

Business Closure

As I noted above, with each passing timestep there is a chance that a business will shut down. Specifically, there are two authored probabilities at play here: a general chance for any business, and a high chance for businesses for which the `demise` year has been reached or eclipsed. Thus, while a shoemaker shop has a `demise` year of 1900, it will not immediately go out of business at that time, but rather the chance of that happening is greatly increased (making it a matter of time before the closure is probabilistically triggered). Likewise, as I explained above, a business will not come to exist right at the time of its specified `advent`, but instead it becomes possible then. In this way, the modeling of the shifting of historical eras with regard to business operations is smooth in its expression across generated towns. When a business closes down, its employees are all laid off (and thus become unemployed) and its building is demolished (by contracting a construction firm in town).

9.2.8 Real Estate

In *Talk of the Town*, characters live in houses or apartment units. The decision making that drives the simulation of real estate works according to the kinds of utility-based procedures that I have outlined in the previous section. Specifically, a character who seeks housing will contract a realtor and then score every vacant home and lot in the town according to a number of concerns. This scoring procedure considers the distance of that home or lot to every other character in the town, with the score increasing when friends are close and either increasing or decreasing when family is close, depending on the character's personality (conscientiousness boosts the score, openness lowers it). In the latter case, the specific familial relation with a relative in town also modulates the score, making conscientious characters more likely to live by a mother than an aunt, for example. Additionally, this scoring procedure models characters preferring homes and lots that are near their workplaces and preferring vacant homes over vacant lots (since the latter require an additional construction step). If a character is married, her spouse will also contribute to the score of a given home or lot.

Once the scoring has been carried out, one of the top three candidates is selected probabilistically. If a vacant lot is selected, then an architect will be solicited and a construction event will be triggered, just like with business construction. Finally, the character—along with her immediate family, if applicable—moves into her new home.

9.2.9 Education

As I noted in Section 9.2.4, children in *Talk of the Town* go to school during the day, assuming there is one in town. Specifically, this will be a K–12 school, which means every student attends the same school.[31] Additionally, there is a chance of a character dropping out of high school to begin work—this depends on the character being offered a job at that time (and also on a set probability, so that not all characters drop out when offered a job). In some humorous cases encountered during *Bad News* performances, students have dropped out to become teachers at the same schools they stopped attending. Earlier, I alluded to the modeling of higher education in *Talk of the Town*, which I mentioned as being very abstract.

[31] In the United States, a 'K–12' school is one that is attended by children of all ages, from kindergartners to high-school seniors. For obvious reasons, they are typically encountered in very small towns.

If an adult character is unemployed in the town for at least four consecutive years, a probabilistic rule may trigger to cause her to instantaneously receive a college degree. This may be viewed as an abstract modeling of the character taking classes at a nearby university outside the town (or by correspondence, even).

9.2.10 Departing the Town

A character may leave her town, and thereby the simulation. Such a departure is permanent, and may be probabilistically triggered for a number of reasons: a child becomes an orphan (this is an absolute trigger); a character is unemployed for a prolonged period; a new adult has a high openness-to-experience personality component, and decides to leave for greener pastures; there are no vacant homes or lots in town; a character simply elects to depart for unspecified reasons (i.e., a general probabilistic rule triggers). When a working character leaves the town, her position is vacated and then filled according to the procedure outlined above. If a married character departs, her family will join her in leaving.

9.2.11 Artifacts

As I explained in Section 9.1.3, there are two kinds of artifacts that are modeled in *Talk of the Town*: *gravestones* and *wedding rings*. When a character dies, a gravestone is generated following the procedure outlined above. As part of the abstract modeling of a marriage event, both spouses begin wearing wedding rings; upon divorce or widowing, a character stops wearing a wedding ring. Unlike in *World*, artifacts do not weather over time.

Artifacts and Knowledge Transmission

As I have noted already, artifacts hook into the system's rich modeling of character knowledge phenomena. Specifically, when an artifact is examined by a character, it transmits information about the subject of the artifact according to a procedure that I will explain in the next section. By examining a gravestone, a character learns the following about the deceased: first name, middle name, last name, suffix, surname ethnicity, surname composition (hyphenated or not), birth year, death year, approximate age (assumes birthday already occurred in death year), condition (that the deceased is dead), and vocation (in the case that an occupational inscription is on the stone). In noticing a wedding ring on a character's finger, an observer will learn that that character is married.

9.2.12 Character Knowledge

Characters in *Talk of the Town* construct (and mutate) subjective knowledge about the storyworld as they go about their daily lives; this rich modeling of knowledge phenomena is probably the hallmark of the system.[32] In this section, I will describe *Talk of the Town*'s modeling and simulation of character knowledge phenomena, including the mechanisms by which knowledge may originate, propagate, deteriorate, and terminate according to the procedures of the architecture.

[32]The subsystem described in this section was designed in collaboration with Adam Summerville and has also been presented in a workshop paper [1061] and a book chapter [1044]. The prose here is a revision of that of the latter document, though it does include some additional information pertaining to updates to the subsystem that were made after the composition of those earlier publications. As such, this section provides the most authoritative reporting on this aspect of the project.

Overview

People and places in a simulated town have perceptible attributes that characters may observe to form corresponding beliefs about those subjects. Such knowledge may then propagate across characters during social interactions. Critically, character knowledge can also be misremembered (in multiple interesting ways), or it may be altogether forgotten. All throughout its modeling of such phenomena, this module keeps track of belief histories and knowledge trajectories, which allows for the tracing of any piece of modeled information.

Perceptible Attributes

Talk of the Town characters have *perceptible attributes*, meaning attributes that are directly observable by other characters. These are mainly physical features, like hair or eye color, but the system also models conditionally perceptible attributes—for instance, a character's workplace is observable while she is in the act of working. A *radius of perceptibility*, in which character attributes may be observed directly, is defined simply as a location in the town (either a dwelling place or a business); thus, characters whose routines have brought them to the same place on a timestep may observe one another at that time.

While a variety of modeled attributes may be perceived—I will list these shortly—perception (along with other knowledge phenomena) is triggered probabilistically according to how *salient* the subject and the attribute are to a given character. As I explained in Section 9.2.5, the system tracks how salient each character is to every character, according to a number of social concerns. Similarly, this subsystem utilizes specified *attribute saliences*, which prescribe how likely given character features are to be observed and to be talked about among characters. In the procedural content that I have authored, this captures, for instance, the real-world phenomenon of features of a person's hair and eyes being more salient than those of her nose and chin, as J. Kirkland Reynolds and Kathy Pezdek [996] and Marcos Ruiz-Soler and Francesc S. Beltran [1032] have reported.

Character saliences work together with attribute saliences to determine the probability of knowledge phenomena occurring for a given character and attribute—that is, the probability of a perceptible attribute being observed, as well as the probability of a belief about any attribute being propagated, misremembered, or forgotten. Below, I will explain this extended salience computation in more depth.

Network Structure

A character's composite knowledge of the world is structured as a *network* of interlinked mental models that each pertain to a single person or location. As Figure 9.4 illustrates, the interlinking occurs when a character's knowledge as to some attribute of a character or place resolves to some other character or place for whom or which they have another mental model. For instance, a character may believe that some person works at some business in town, and so her belief about that person's workplace would itself link to her mental model of that business (as I explain below). This network structure provides both elegance and convenience, since it allows characters to reason about entities in terms of knowledge they may already have about related entities (rather than by instantiating redundant or potentially inconsistent knowledge). For example, if a character knows where another character works, she may reason about the latter's work address in terms of existing knowledge about the workplace that can be stored and accessed independently of knowledge about the character who works there.

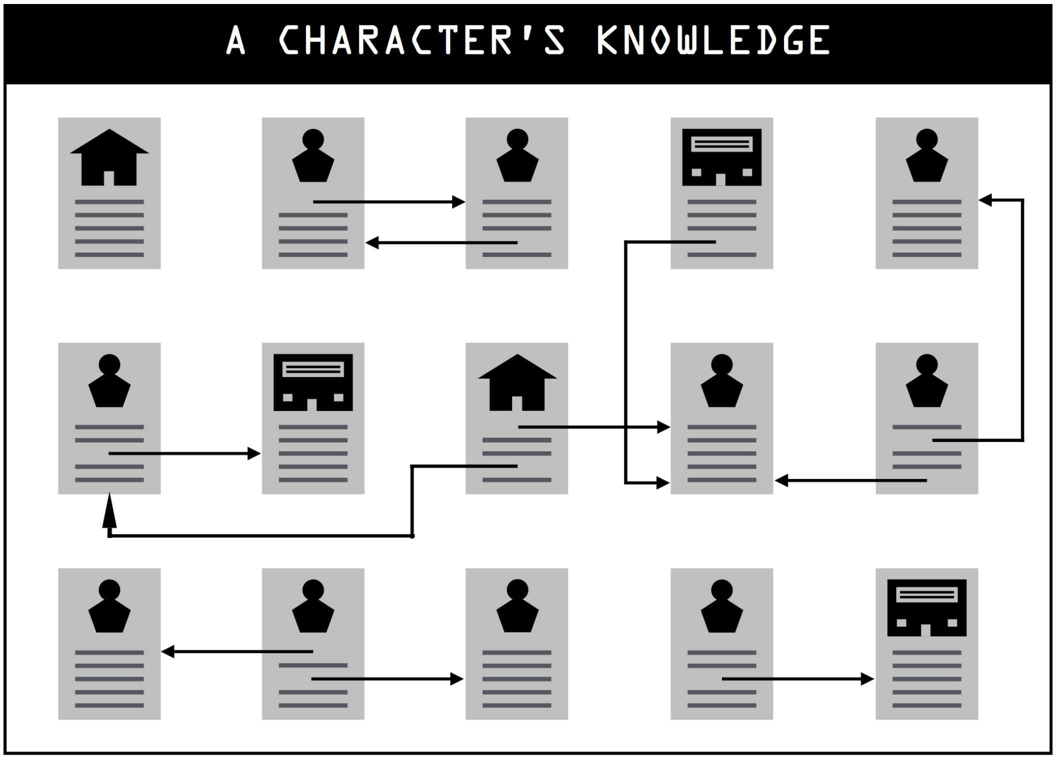


Figure 9.4: A *Talk of the Town* character's composite knowledge of her storyworld as an interlinked network of mental models. Over the course of simulated life, a character builds up knowledge about the characters and locales around her by maintaining *mental models* about each that integrate subjective *belief facets* about the various perceptible attributes of that subject. In total, a character's composite knowledge of her storyworld takes the form of a *network* that interlinks mental models whenever a facet of one resolves to a pointer to another entity. For example, if a character believes that someone works at a particular town business, then the `workplace` belief facet in her mental model for the character will point to her mental model for the business. Note: this illustration is simplified—an actual character's composite knowledge will integrate beliefs about hundreds or thousands of entities.

Mental Models

As I have already noted, characters form *mental models* about town residents and landmarks. Following *PsychSim* [746, p. 243] and *Thespian* [1152, p. 23], two earlier systems in which characters also richly perceive one another, I selected this term specifically to indicate that the structure of a character's knowledge of some entity closely matches the structure of how that entity is represented in the simulation itself. That is, the components of a mental model generally correspond to the components by which the simulation actual models a character, the set of which I outlined in Section 9.1.2. Each of a character's mental models pertains to a specific individual entity (either a character or a home or a business) and is composed by *belief facets* that each pertain to an individual attribute of that entity. In total, there are forty facet types that are modeled in the version of *Talk of the Town* that is used to drive *Bad News* (these are grouped conceptually for presentation clarity):

- For mental models of characters:
  - **Status.** Condition (`alive` or `dead`), year of departure from the town (if any; e.g., `1972`), and marital status (`single`, `married`, `divorced`, or `widowed`).
  - **Age.** Birth year (e.g., `1941`), death year (if any), and approximate age (e.g., `30s`).
  - **Name.** First name, middle name, last name, surname ethnicity (e.g., `German`), surname construction (whether it is hyphenated), and suffix (e.g., `III`).
  - **Appearance.** Each of the twenty-four facial attributes that were listed in Section 9.1.2.
  - **Occupation.** Workplace (links to mental model of that place), vocation (e.g., `tailor`), shift (`day` or `night`), and status (`employed`, `unemployed`, or `retired`).
  - **Home.** Home (either an apartment unit or a house; links to mental model of that place).

- **Whereabouts.** Where a person was on a given timestep (links to mental model of that place). For example, a character could believe that someone was at `Horswill Farm Park` or the `Law Offices of Mateas and Wardrip-Fruin` on the night of August 12th, 1979. Such whereabouts knowledge was critical in the design of *Talk of the Town* (2015), since these beliefs would pertain to where characters were on the night that the lover was witnessed visiting the late town founder.

- For mental models of businesses:
  - **Employees/Residents.** A list of its employees (each links to mental model of a character).
  - **Block.** Business block, for example, `800 block of Lessard Avenue`.
  - **Address.** Business address, for instance, `222 Elson Street`.
- For mental models of homes:
  - **Residents.** A list of its residents (each links to mental model of a character).
  - **Apartment.** Whether it is an apartment unit (`yes` or `no`).
  - **Block.** Home block, for example, `300 block of Lyndale Avenue`.
  - **Address.** Home address, for instance, `613 Fillmore Street`.

Each facet is structured as a collection of data about the belief. In addition to its *owner* (who has constructed the mental model), *subject* (the character to whom it pertains), and *facet type* (e.g., `first name`), this data includes:

- **Value.** A representation of the belief itself, for example, the string `brown` for a belief facet pertaining to hair color.

- **Mental model.** If the value of this facet resolves to an entity for whom the owner of this facet has formed a mental model, this component will point to that mental model.

- **Predecessor.** The belief facet that the owner previously held, if any. This allows the system to track supplanted or forgotten character knowledge (about which more soon).

- **Evidence.** A list of the pieces of evidence by which the owner of this facet formed and continues to substantiate it. As I explain in the next section, all character knowledge is undergirded by evidence that a character accumulates as the simulation proceeds.

- **Parents.** If this knowledge originated in information from other characters, this will point to the belief facets, owned by those characters, that spawned this current facet. Specifically, if a piece of evidence represents that a character learned information from someone else, this component will point to that character's corresponding belief facet. This allows the module to trace the history and trajectory of any piece of information.

- **Strength.** The strength of this particular belief. This is the sum of the strength of all pieces of evidence supporting this belief, whose determination I will explain in the next section.

- **Accuracy.** Whether or not the belief is accurate (with regard to the *current* true state of the world). Thus, the accuracy of a belief can change over time as the storyworld changes. For example, if someone observes another character at work, a corresponding work-status belief facet (set to `employed`) will be accurate; but if later on the working character retires without the observer finding out, the belief facet will have become inaccurate due to the storyworld changing in the interim.

Evidence

All character knowledge is formed in response to evidence, and knowledge may also propagate, deteriorate, or terminate in a way that can be described using pieces of evidence. I will illustrate these details by way of an *evidence typology*, which comprises twelve *types* across five conceptual categories.[33]

- How knowledge *originates*:
  - **Reflection.** A *reflection* occurs when a character perceives something about herself, which happens at every timestep.
  - **Observation.** When a character directly *observes* a person or place, she may form knowledge about attributes of that entity. Whether she forms knowledge about a particular attribute depends on the salience of the entity and the attribute type, as I will explain shortly.
  - **Examination.** When a character *examines* an artifact (either a gravestone or wedding ring), she forms knowledge about its subject entity (either the deceased of the ring wearer). As I explained in Section 9.2.11, a wedding ring imparts that a character is married and gravestones express the following information about the deceased subject: first name, middle name, last name, suffix, surname ethnicity, surname composition (hyphenated or not), birth year, death year, approximate age (assumes birthday already occurred in death year), condition (that

[33] This typology has evolved since the earlier publications that reported on this subsystem [1061, 1044], and as such, this section provides the only explanation of the final state of the system with regard to these concerns. Most pertinently, the simulation of artifacts transmitting knowledge is presented here for the first time.

deceased is dead), and vocation (in the case that an occupational inscription is on the stone).

- **Transference.** If one entity reminds a character of another entity (determined by feature overlap between her respective mental models of them), she may unconsciously *transfer* a belief she holds about one so that its corresponding facet value is also attributed to a facet in her mental model of the other entity.

- **Confabulation.** By *confabulation*, a character *unintentionally* concocts new knowledge about some entity. The particular facet value that gets confabulated is determined probabilistically according to the distribution of that feature type in the confabulator's town. Thus, in a town with a predominance of brown hair, a character would be most likely to confabulate someone as having that feature value.

- **Lie.** A *lie* occurs when a character *intentionally* conveys information to another character that she herself does not believe. This is deemed a type of origination (and not propagation) because the knowledge in question is *invented* by virtue of the lie—that is, it does not propagate any existing knowledge.

- **Implant.** As I mentioned in Section 9.2.2, for reasons pertaining to computational efficiency, the system does not simulate character knowledge phenomena until the end of world generation. At the beginning of this transition into a higher granularity of simulation, essential character knowledge is directly *implanted* into character minds, and thus will have no explicit point of origination. I will discuss this notion of knowledge implantation in depth below.

- How knowledge *propagates*:

- **Statement.** A *statement* occurs when a character conveys information to another character that she herself believes. Whether characters will exchange a particular piece of information depends on the salience of its subject and type, as I will discuss shortly.
- **Eavesdropping.** Characters may *eavesdrop* on nearby social interactions to overhear statements and lies; this happens at a set probability.

• How knowledge *reinforces itself*:

- **Declaration.** Whenever a character delivers a statement, the strength of her own belief (being *declared* by her statement) will slightly increase. That is, the more a person retells some belief, the stronger that belief becomes, which is realistic [1342]. By this mechanic, a character who frequently tells the same lie might come to actually believe it.

• How knowledge *deteriorates*:

- **Mutation.** As an operationalization of memory fallibility, knowledge may *mutate* over time. This is modulated by a character's memory attribute (see Section 9.1.2) and the facet type (e.g., a whereabouts belief will be more likely to mutate than a belief about someone's first name). The particular mutation that occurs is determined by the *belief-mutation graph*: an authored schema that specifies state-change probabilities given a facet value. For instance, given the value `brown` for a facet about someone's hair color, the module will consult this graph to decide how to probabilistically mutate `brown`—here, the graph captures that, for example, `black` and `red` are more probable mutations than `white` or `purple`. Figure 9.5 illustrates an excerpt from the graph that I authored for *Talk of the Town*.

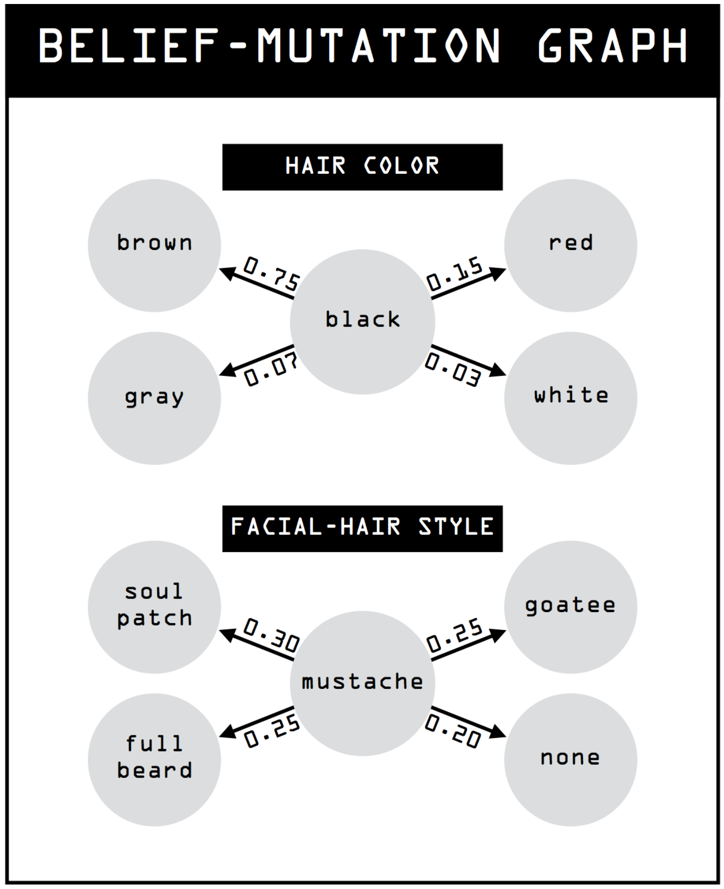


Figure 9.5: An illustration of an excerpt from the *belief-mutation graph* that I authored for *Talk of the Town.* This schema specifies state-change probabilities by which a particular belief about some entity might mutate into a different belief. In this excerpt of the graph, for example, there is a 75% chance that a `black` belief-facet value pertaining to someone's hair color will mutate to `brown` and a 3% chance that it will mutate to `white`. For certain facet types, mutation cannot be captured using state transitions, and so in these cases I defined functions that take current facet values as their arguments and returned mutated ones—for example, mutation of an address belief may modify the street name according to an adjacent street. Because there are forty facet types modeled by the system, the authorial burden of specifying all the probabilistic mutation was immense.

- How knowledge *terminates*:
  - **Forgetting.** To further incorporate memory fallibility, knowledge may be *forgotten* due to the passing of time. This is affected by a character's

memory attribute and the salience of the facet subject and type.

Characters are not consciously aware of transferences, confabulations, or mutations, and recipients (as well as eavesdroppers) of lies treat them as statements. That is, the recipient will reason about a lie as if it were a statement (and so the strength of a lie, as a piece of evidence, is equal to that of a statement), but the system nonetheless tracks that it was in fact a lie (to allow for reasoning about true knowledge trajectories). Additionally, each piece of evidence has metadata of the following types:

- **Source.** In the case of a statement, lie, or eavesdropping, a piece of evidence's *source* is the character who delivered the information. This metadata allows the system to trace the history and trajectory of any piece of information, which was a major design goal in developing this project. With regard to *Talk of the Town* (2015), my collaborator Adam Summerville and I envisioned the trajectories of important pieces of knowledge (ones that affected the outcome of the game) being presented to the players at the end of the gameplay session. For example, if a non-player character told a player that the lover has a mustache and the player then identified a mustached character as being the lover, the system could visualize how that information originated in a critical lie constructed by the lover character.

- **Location.** Where the piece of evidence originated (e.g., the specific location where an observation or statement took place).

- **Timestep.** The timestep on which a piece of evidence originated.

- **Strength.** The *strength* of a piece of evidence is a floating-point value that is determined by its type—for example, a mutation is weaker than an observation—and decays as time passes. In the case of statements, lies, and eavesdroppings, the strength of a piece of evidence is also affected by the affinity its owner has for its source and the strength of that source's own belief at the time of propagation.

Salience Computation

When a character observes some entity in the simulation, a procedure is enacted that determines, for each perceptible attribute of the observed entity, the probability that the character will remember what she saw; this procedure crucially depends on the salience of the entity and attribute being observed. Salience computation considers the relationship of an observed character (subject) to the observer (e.g., a co-worker is more salient than a stranger), the extent of the observer's friendship with the subject, the strength of the observer's romantic feelings toward the subject, and finally the subject's occupational status (characters with more prestigious job positions are treated as more salient).

For locations, salience computation currently only considers whether the observing character lives or works at the observed place. Additionally, this salience-computation procedures consult an authored schema specifying the salience of each attribute type. My authored knowledgebase captures, for instance, that features of a person's hair and eyes are more salient than those of her nose and chin [996, 1032]. Salience computation is also used to determine the probability that a character will misremember or altogether forget (on some later timestep) knowledge pertaining to some subject and attribute.

Knowledge Implantation

The simulation of character knowledge phenomena is computationally expensive. If all of these phenomena were to be simulated during the entire period of world generation—the town's total history—it could take several hours to produce a town. Moreover, a lot of computation would be devoted to modeling the knowledge of hundreds of characters who would have died long before the period of *Bad News* gameplay. In the case of that experience, a town must be generated in a few minutes or less, because we schedule performances one after the other to afford maximum player throughput.[34]

Thus, in *Bad News*, this simulation detail is not worth the time and computation, and so the system approximates the history of character knowledge phenomena using a procedure of *knowledge implantation*. By this pattern, world generation employs all aspects of the simulation besides ones related to character knowledge—characters still form relationships, start businesses, and so forth—and then terminates one week prior to the specified end date in the summer of 1979. At this point, however, living characters have no knowledge at all. To resolve this, the module employs a procedure that *implants* into each character's mind the knowledge that would believably be ingrained in them. The following pseudocode illustrates this procedure:

```
for resident in town
    implants = []
    for immediate family member of resident
        add immediate family member to implants
    for friend of resident
        add friend to implants
    for neighbor of resident
        add neighbor to implants
```

[34] Additionally, it would not be feasible to generate a number of towns at the beginning of a day of performances, because *Talk of the Town* towns take up a considerable memory (on the other of several gigabytes each).

```
for coworker of resident
    add coworker to implants
for every other character who has ever lived
    chance = 1.0 − (1.0/salience of that character)
    if random number < chance
        add other character to implants
for character in implants
    for attribute of character
        chance = attribute salience
        chance += −1.0/salience of that character
        if random number < chance
            have resident adopt accurate belief
```

Knowledge Propagation

When two characters engage in a social interaction (see Section 9.2.5), they may exchange knowledge about other characters and locations in the town—these entities about which knowledge is exchanged are called the *subjects of conversation*. Currently, the subjects of conversation that come up in an interaction between two characters is determined by computing the total salience of all the entities that either of the characters know about. The n highest-scoring entities are then brought up in conversation, with n being determined by the strength of the interactors' relationship and also their respective *extroversion* personality components (each of these work to increase n).

For each subject of conversation, the characters exchange information about individual attributes of that subject (corresponding to the individual belief facets of the characters' mental models of that subject) according to the salience of each attribute type. Because a character may bring up subjects that her interlocutor does not (yet) know about, this propagation mechanism allows characters to learn about other people and landmarks that they have never themselves encountered. It is even possible for a character to learn about another character who died before she was born; this often occurs when parents tell their children about deceased relatives (who score highly in salience computation due to being related to both conversational partners).

Lies

During a social interaction, a character may convey false information about the subject of conversation—more precisely, information that she herself does not believe—to her interlocutor. Currently, this is triggered probabilistically according to a character's affinity toward the interlocutor, and the misinformation is randomly chosen. While I had a number of plans for extending this aspect of the system [1061, p. 60], I never got to them.

Fallibility Modeling

As an operationalization of *memory fallibility*, characters may adopt false beliefs for reasons other than lies. On each timestep after world generation, the four phenomena associated with memory fallibility—transference, confabulation, mutation, and forgetting—are probabilistically triggered for character mental models. When this happens, the facets of that mental model are targeted for deterioration at probabilities determined by the character's memory attribute (again, modeled as a floating-point value), the facet type (e.g., a whereabouts belief will be more likely to mutate than a belief about a first name), and the strength of the existing belief (weaker beliefs are more likely to deteriorate).

As I explained already, mutation works according to the belief-mutation graph and a series of handcrafted procedures for certain other types. As another example of the latter, if a character will misremember another person's surname, the procedure retrieves a surname of the same ethnicity (from the surnames corpus) that is similar to the source name (modeled using edit distance between the strings representing each).

Belief Revision

Over time, as the storyworld changes and information travels and beliefs warp and mutate, characters carry out procedures of *belief revision*. A character always forms a belief upon encountering a first piece of evidence supporting it, assuming she has no current belief that it would replace. As a character accumulates further evidence supporting her belief, its strength will increase commensurately to the strength of the new evidence. Additionally, whenever a character delivers a statement, the strength of her own belief (that she is imparting with the statement) will slightly increase. That is, the more a person retells some belief, the stronger that belief becomes for her, which is realistic [1342]. If that character, however, encounters new evidence that contradicts her currently held view, she will consider the strength of the new evidence relative to the strength of her current belief. If the new evidence is stronger, she will adopt the new belief that it supports; if it is weaker, she will *not* adopt a new belief, but will still keep track of the other *candidate belief* and the evidence for it that she had encountered. If she continues to encounter evidence supporting the alternative candidate belief, she will update its strength accordingly and, if at any time that strength exceeds the strength of her current belief, she will adopt the candidate belief and relegate the previously held belief to candidate status.

As an example, let us consider a hypothetical scenario involving two characters named Michael and Noah. Michael has just gotten a haircut, but Noah still believes that Michael has long hair because he had seen this for himself on multiple occasions and heard the same from a good friend, Jim, just yesterday. During a conversation, another character that Noah does not know very well mentions to him that Michael has short hair. Noah does not immediately change his belief about Michael's hair, because the strength of this new evidence does not outweigh the strength of his current belief (since he had heard only yesterday that Michael has long hair and moreover saw this himself on multiple occasions). Later, Noah's wife tells him that Michael's hair is in fact short. Noah now adopts the new belief, because the strength of the evidence constituted by both of these statements now outweighs the strength of the evidence supporting his prior belief. The prior belief was supported by more pieces of evidence, but this evidence was so outdated relative to the new information that its strength had decayed to the point where the new evidence was now stronger. Critically, this process of belief revision depended on Noah keeping track of the candidate belief—that Michael has long hair—even during the period prior to him adopting that belief.

Related Work

While story generators and expressive multiagent systems have typically featured characters who operate over perfect knowledge of the world—for example, see [51, 333]—a fair amount of systems have incorporated models of character belief. As I noted above, characters in Sheldon Klein's murder-mystery generator can recursively simulate the storyworld in accordance with their subjective beliefs [597, pp. 29–30]. In *Tale-Spin* [822], discussed at length in Chapter 4, characters may be initialized to have some knowledge about the storyworld and may perceive which characters and objects are nearby. Agents in applications of the Oz Project similarly perceive their surroundings, but at a higher fidelity [86]. This is likewise seen in Hans ten Brinke, Jeroen Linssen, and Mariët Theune's (2014) extension [1244] to the *Virtual Storyteller* system [1230], in which updates to character beliefs are fed to a reactive narrative-planning system.

Jonathan Teutenberg and Julie Porteous have also explored narrative planning with perceptive characters [1245]. Stories generated by *Minstrel* [1276] may rely on false character beliefs, but such circumstances must be artfully crafted beforehand [937]. Chris Crawford has written extensively on an approach for modeling character gossip [224, pp. 235–244], including lies [222]. In a project that I mentioned earlier, Pablo Gervás has used chess gameplay data to generate stories focalized to individual pieces with limited *fields of perception* [388]. In the emergent-narrative system of David Carvalho and collaborators [168], characters may operate over false beliefs. Recent work by Alireza Shirvani, Rachelyn Farrell, and Stephen G. Ware represents character beliefs in terms of possible worlds [1147, 1146]. In *Tale-Spin*, *PsychSim* [746], *Thespian* [1152], *Othello* [175], and Henrique Reis's extension [995] to the *FAtiMA* agent architecture [51], characters not only perceive the world, but form mental models of other characters and *their* beliefs—as such, these systems operationalize *theory of mind* [362].

A handful of multiagent systems have specifically explored *deception in agents* [1309], a notion that naturally relies on the modeling of knowledge. *Tale-Spin* characters may lie to one another in pursuit of their goals [822, pp. 183–184]. *Golem* implements a *blocks world* [64] variant in which agents deceive others to achieve goals [331], while *Mouth of Truth* [248] uses a probabilistic representation of character belief to fuel agent deception in a variant of Turing's *imitation game* [1274]. In David Christian's excellent masters thesis [186], he reports on a *deception planner* that injects inaccurate world state into the beliefs of a target agent so that she may unwittingly carry out actions that fulfill ulterior goals of a deceiving agent. Agents in Reis's aforementioned extension to *FAtiMA* [995] employ multiple levels of theory of mind to deceive one another in the party game *Werewolf*

[393]. Matthias Rehm and Elisabeth André describe a system in which an embodied agent lies during a game of dice, but exhibits realistic facial cues while doing so [989]. Henry Mohr, Markus Eger, and Chris Martens recently developed a system that generates murder mysteries surrounding characters who form knowledge and may hide the truth [843]. Eger's *Impulse* language [301] for modeling stories explicitly represents character mental models, and his recent dissertation dives deep into the modeling of agent beliefs (and theory of mind) in games that center on player knowledge [300]. While all of the above systems showcase characters who perceive—and, in the latter cases, deceive—other characters, all of them appear to place less emphasis, relative to *Talk of the Town*, on knowledge propagation and memory fallibility.

Outside of *Dwarf Fortress* [17], which I discuss next, commercial and even research videogames have been surprisingly spare in their modeling of character knowledge. There are many examples of stealth and action games that use complicated knowledge, perception, and alertness models to produce short-term non-player character (NPC) beliefs that are a core part of gameplay [265, 1325, 1306]—here, notable examples include *Thief: The Dark Project* [677] and *Third Eye Crime* [529]. Few projects, however, have supported characters whose perceptual systems instantiate memories or lasting beliefs—though this has certainly been proposed [530]—and there are even fewer examples of the modeling of fallible character memory. This is curious, since issues of belief, especially false belief, are often central in other fictional media [913]. Some games *are* about character beliefs, to be fair, but in these cases beliefs are typically handcrafted, as in the AAA game *LA Noire* [1243].

In games that do model character knowledge procedurally, the AI architecture that handles such concerns is sometimes called a *gossip system*. A classic example of this type of module is the reputation system in *Neverwinter Nights* [141]. More generally, *reputation systems* [432] (or *faction systems* [539]), which alter NPC behavior toward the player character according to actions she has taken, are a form of character knowledge modeling, but a rudimentary one in which NPCs seem to globally inherit perfect knowledge of player behavior. This can play out quite awkwardly, as in the familiar case in which NPCs in one area of the gameworld appear cognizant of a player's crime in another area before word of it could have ever spread to them [978].

Phil Goetz's forgotten application of *SNePS* [1142] to the domain of interactive fiction explored the development of NPCs who harbor subjective models of the simulated world [398]. *Black and White*'s "creatures" were special NPCs that could acquire knowledge (by *reinforcement learning* [1224]) about how to do certain tasks [322]. Interestingly for a mainstream title, the recent *Middle-earth: Shadow of Mordor* [845] features NPC enemies who may reference earlier combat [620, 247]. More commonly, games utilize simple *flags* that mark what plot knowledge the player currently has, so that only valid dialogue (sub)trees are deployed [1311]. By extending *Comme il Faut* [804]—the AI system underpinning *Prom Week*, which features omniscient characters [803]—*Mismanor* [1211, 1212] likewise incorporates a flag-like representation of player plot knowledge, but also a more general notion of character world knowledge (which specifies things like characters' knowledge of other characters' traits, something that *The Sims 3* also tracks [324]). Most interestingly, the player may manipulate NPC knowledge by asserting facts about the world, but this information does not propagate because the NPCs do not interact with one another in this same way. Jonathan Rappner has prototyped a system that does model knowledge propagation, at the level of discrete agent interactions [978]. In *Versu*, characters cultivate beliefs that pertain to specific story concerns and, through dialogue interaction, can even persuade other characters to adopt such beliefs [326]. Because *Versu*'s AI framework has only been applied to storyworlds with a small number of characters, it is probably not accurate to say that it features knowledge propagation; moreover, it does not appear to operationalize memory fallibility or character lies. Recently, Sasha Azad and Chris Martens reported on a system in which characters form opinions and exchange viewpoints [56]. Lastly, the *LabLabLab* title *Hammurabi* features characters who may lie to the player in a procedural manner [682].

It seems that *Dwarf Fortress* [17] best represents the state of the art (certainly in the domain of computational media) in modeling character knowledge and its attendant phenomena. In the earlier paper on *Talk of the Town*'s simulation of character knowledge phenomena, I reported a series of insights that were gleaned from personal communication with Tarn Adams (May 28, 2015):

> The primary character-knowledge machinery in *Dwarf Fortress* is its *rumors system*. Rumors typically originate when a character witnesses an important event, such as a crime, and are instantiated at certain local and regional levels. Interestingly, characters in the purview of a rumor might then propagate it to new areas they visit. Examples of this include diplomats informing their home civilizations about traps they have encountered, caravans (who visit player fortresses) bringing rumors from their parent civilizations, and artistic and scientific knowledge moving across civilizations. Like a few other systems noted above, *Dwarf Fortress* also features characters who autonomously lie. When a character commits a crime, she may falsely implicate someone else in a witness report to a sheriff, to protect herself or even to frame an enemy. These witness reports, however, are only seen by the player; characters don't give false witness reports to each other. They may, however, lie about their opinions, for instance, out of fear of repercussions from criticizing a leader. Finally, *Dwarf Fortress* does not currently model issues of memory fallibility—Adams is wary that such phenomena would appear to arise from bugs if not artfully expressed to the player. [1061, p. 61].

Since that reporting, *Dwarf Fortress* has undergone a number of updates, notably one that is centered on storyworld *artifacts* and what characters know about them. As part of this update, characters may now adopt secret identities, as Tarn Adams explains in a recent invited paper on the topic [16].

The approach taken in *Talk of the Town* exhibits several features that are found independently across the various systems that we have outlined. As in systems like *PsychSim* and *Thespian*, characters form mental models of one another. Following *Golem*, *Mouth of Truth*, *Dwarf Fortress*, and other projects, characters may lie to one another, though my system does not model this aspect nearly as richly. Along with *Dwarf Fortress*, *Talk of the Town* appears to be among the few systems to support detailed character knowledge propagation.[35] Lastly, to my knowledge, this system features the richest modeling of memory fallibility to yet appear.

9.3 Emergent Phenomena

As I did with *World* in Section 7.4, I will now discuss examples of intriguing emergent phenomena in *Talk of the Town*. In my view, this engine cannot produce full-fledged narrative sequences because it does not model concrete events beyond abstract social interactions and nondescript life events (such as a house construction), but its modeling is rich enough to produce scenarios that have more to them than *World*'s poetic images. As such, while *World* produces emergent stories and images, as I explained above, *Talk of the Town* generates what may more aptly be described as *emergent scenarios*.

[35] This phenomenon could likely be supported by more of these systems if they were to include more characters in their simulations.

Bar-Crossed Lovers

I recently reached out to my *Bad News* collaborators, Adam Summerville and Ben Samuel, to ask them about their favorite *Talk of the Town* scenarios that emerged in the context of our many performances of the former. Before providing these examples, I should note a few things about *Bad News*, the subject of the next chapter. In this project, a player explores a generated *Talk of the Town* town by interacting with a wizard-of-Oz configuration mounted as an installation-based experience. Each performance begins with the death of a resident of the town, and the player's job is to find and notify the next of kin, which she does by speaking with residents in the town who are all played live by a human actor (Ben).

Here is what Adam had to say about his favorite *Talk of the Town* emergent scenario (citations inserted for clarity):

> My favorite situation to arise from the simulation was something like a Shakespearean tragedy, set in small town America. The deceased was the scion of a family that had run one of the two bars in the town, with a history nearly as long of that of the town itself, and he was in love with the daughter of the owner of the other bar in town. The two bar-owning families had hated each other for generations, but the two children of the families were able to get past that and fall in love. For this to happen, only to have tragedy befall their star-crossed love feels like Bad News was cribbing from one of my favorite movies, "Scotland, PA" [858]—a retelling of Macbeth [1136] in a fast food restaurant in the 1970's. I will always be amazed by the kinds of emergent stories that Bad News is capable of generating.[36]

While the tragical component of this scenario originates at the layer of the *Bad News* experience (since the scion of the first family died in that context), the rest of it emerged out of the various interworkings of *Talk of the Town*'s simulation.

For instance, the *business rivalry* is an emergent of the engine's business simulation, and the *family rivalry* is actually an example of a common system dynamic,

[36] Personal communication, June 14, 2018.

which works as follows. Since characters inherit personality traits, over time a given family in a town will come to be associated with a personality archetype. In turn, due to the social simulation's modeling of personality compatibility, it is possible for one family's archetype to be incompatible with that of another—when this happens, enmities will tend to form between the members of each family. Coupled with the business rivalry (which in this example only required the presence of two bars in the town), this family incompatibility yields the emergent case of a family rivalry. Finally, the romantic attraction between the two characters is a basic phenomenon that is modeled by the system's social simulation, as I explained in Section 9.2.5. When all of these emergent phenomena stack up at a higher order, the result is what I am calling an *emergent scenario*—in this case one with a distinctly Shakespearian quality.[37]

Narrative Cornucopias

The beauty of *Talk of the Town* is that every one of its generated towns overflows with emergent scenarios like the one Adam recalls so fondly. This is because it follows *Dwarf Fortress*'s lead in modeling a storyworld with an abundance of simulated material, resulting in what I have termed (in Section 5.2) a 'cornucopia' of narrative potential—this is the 'overgenerate' component of my curationist tagline *overgenerate and curate*.

[37]Intriguingly, a quite similar emergent scenario underpinned a *Talk of the Town*-driven project that was carried out by researchers and students at the University of Turin in Italy [238]. After collaboratively exploring the generated material of a particular *Talk of the Town* storyworld, students in a seminar run by Rossana Damiano, Vincenzo Lombardo, and Antonio Pizzo homed in on a romance between two characters who worked at rival bakeries. The result of this endeavor, *Hot Bread*, is a work of curationist emergent narrative that hooks into the researchers' *DoPPioGioco* framework for computationally assisted live performance [237]. The following year, I visited Italy to collaborate with this group on what became *Cattive Notizie*, a variant of *Bad News* that I describe in Section 10.2.5.

Deli Drama

Adam's example calls to mind a related scenario that concerned a character who dropped out of high school to start a delicatessen that would rival one that already existed in the town. This alone is mildly interesting, but in this particular case the rival deli was owned by the character's own mother. Due to having incompatible personalities, the mother and son did not like each other, and so they did not speak during the five years following the son's opening of the rival establishment. Eventually, this situation would also turn tragic, because the son was in fact the deceased character in the *Bad News* performance for which the town was generated—as such, the mother was the next of kin, and so this performance culminated in an emotional scene in which she (played by Ben Samuel) realized that she would never have a chance to rectify the unfortunate family situation. Many towns feature a more common type of family drama in the form of a *sibling rivalry*: two siblings with a mutual enmity.

The Recluse with Green Hair

While the richest scenarios combine material generated by various subsystems, just as Adam's example does, I have encountered instances pertaining to only one subsystem that are still narratively potent. One notable example pertains to a dynamic of the knowledge subsystem, whereby schools become the hot spots for gossip and popular kids in turn operate as the primary manipulators of knowledge in the towns. This may be unintuitive, so let me explain. While adults generally go off to various workplaces during the day, every child between five and eighteen years old spends such timesteps at a single K–12 school. As such, children have more opportunities for social interaction and knowledge propagation—since there are more nearby characters—and so a town's K–12 school tends to become a nexus for character knowledge. This pattern is further driven by children eavesdropping on their parents' social interactions at night and then going into school during the day to spread the information obtained thereby. Moreover, because the children with the most friends will engage in the most social interactions, popular kids become the primary drivers of (mis)information.

In one hilarious scenario, an introverted hermit who few had ever met became the subject of a widespread misbelief that his hair was green—upon investigation, I discovered that this false gossip had originated with a confabulation by the town's most popular kid. As such, the basic dynamics of character routines and character knowledge phenomena had produced a humorous variation on the literary trope of the *misunderstood recluse* (e.g., as exemplified by Harper Lee's Boo Radley [671]).

The Night Beat

Another favorite scenario of mine emerged out of the basic dynamics of *Talk of the Town*'s affinity modeling. This situation concerned two police officers who worked the night shift in a town. One of these officers was a man who was among the town's most beloved individuals—he was maximally extroverted and maximally agreeable, which means he selected friends often and was selected as a friend frequently. The other officer was essentially his opposite: she was maximally introverted and maximally disagreeable, which led to her disliking nearly everyone she met and being considered a friend by few—she was a *misanthrope*.

However, due to the male officer's extreme agreeableness and especially to the amount of time the two spent together, he came to consider her his best friend; in turn, due to the amount of time spent together, she considered him her worst enemy. This kind of asymmetry is fairly common in the system, but in this case it was also stacked atop a case of emergent romantic intrigue: even though she disliked him as a friend, the woman was madly in love with the male officer; in turn, the man was extremely romantically averse to the female officer. As such, in total this scenario featured three kinds of interesting asymmetry: platonic asymmetry, romantic asymmetry (i.e., *unrequited love*), and internal asymmetry (each character simultaneously harbored asymmetrically extreme feelings for the other). The result, framed in the peculiar context of a small town's night beat, felt like a set-up for a *buddy cop* [1278] romantic comedy.

Yvette and the Love Polygon

When it comes to romantic affinity, a classic kind of emergent intrigue is the *love triangle* dynamic [1279]. In popular media, this trope can take a number of forms, but generally it involves three characters who are each in love with one of the others, who in turn does not reciprocate. At its most complex, a love triangle involves a character A who only loves a character B, who in turn only loves a character C, who in turn only loves A.

At the Workshop on Tutorials in Intelligent Narrative Technologies, held in Los Angeles in 2016, I led an interactive exploration of a *Talk of the Town* town that centered on a randomly selected character named Yvette. Even though the character was chosen at random, her life situation had no scarcity of social intrigue: Yvette was a teenage girl in 1979 who was in love with her stepsister Lori, who was married to David but was in love with David's brother Brian, who loved Lori back, while in turn David was himself in love with Yvette (who was thus his wife's stepsister). During the interactive session, Kate Compton took to a whiteboard to map out the welter for the benefit of all our understanding—Figure 9.6 shows an excerpt from this diagram.[38]

[38] In the session, we followed Yvette over the course of her life, from her teenage years in the 1970s to the early 2000s, by simulating five years at a time. Eventually, she fell for other

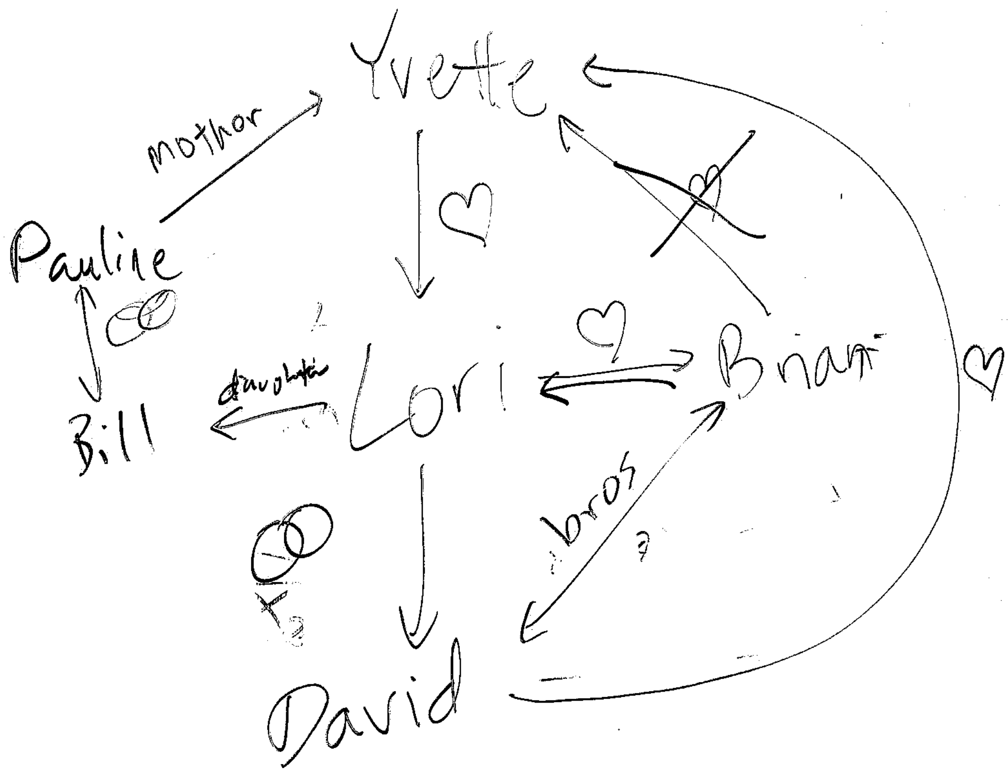


Figure 9.6: An illustration of emergent social intrigue in *Talk of the Town*. This spontaneous whiteboard diagram by Kate Compton maps out the social dynamics surrounding a teenage character named Yvette. The annotations on the arrows indicate romantic infatuation (heart), marriage (interlocked circles), and familial relationships ('mother', 'daughter', 'bro[ther]s'). As the diagram expresses, Yvette is in love with her stepsister Lori, who is married to David but is in love with David's brother Brian, who loves Lori back, while in turn David is himself in love with Yvette (who is thus his wife's stepsister). Such emergent *love polygons* are ubiquitous in *Talk of the Town*'s generated social networks, and they emerge from a system of simple social mechanics.

In *Talk of the Town*, such dramatically potent structures appear in every town, which again I attribute to the utilization of an 'overgeneration' approach to simulation that produces narrative cornucopias. More often than love triangles, towns

women who were not Lori, but tragically there was no one else in town with whom there was a mutual attraction. The workshop attendees collectively wished for her departure from the town with each passing half decade, but sadly she stuck around her entire life, with all her various infatuations going unrequited.

feature *love polygons*, like the one that Yvette anchored. Sometimes these chains of unreciprocated love may span upward of dozens of characters—this may seem unrealistic, but it is hard to say for sure, since we do not have access to every human's true feelings in the real world. As such, this phenomenon may be like the unintuitive but realistic genealogical structures that emerge in *World*. In that case, and potentially here as well, such complex structures are only discernible from the omniscient perspective afforded by computer simulation.

The Interpersonal Circumplex

Another basic emergent social dynamic concerns the juxtaposition of character affinities and character *status* relations [498, 550], which produces the *interpersonal circumplex* [1338], a model of interpersonal relations that maps affinity against patterns of dominance and submission. In *Talk of the Town*, there are a number of reified status relations: boss–employee, senior employee–junior employee, elder family member–younger family member, and so forth. While the interpersonal circumplex is not explicitly modeled in the social simulation, its various expressions could be recognized by a second-order system that processes its emergent material. Indeed, we do this in *Bad News*, where affinity and status—the latter of which is a central concern in improv [550]—are primary concerns for Ben as he brings *Talk of the Town* characters to life. Some researchers have argued that the circumplex's affinity and status axes are actually equivalent to the agreeableness and extroversion components of the five-factor model of personality [809]—of course, the various dynamics yielded by this notion could also be easily recognized in *Talk of the Town*'s emergent material.

The Restaurateur

I should also note that much of *Talk of the Town*'s emergent intrigue is the byproduct of a mere *improbability*—that is, situations or occurrences that are interesting simply because they are improbable. Such intrigue characterizes, for instance, the otherwise benign matter of a character surviving an extensive number of potentially lethal dice throws (by receiving unlikely outputs from the random number generator). I have already mentioned the 111-year-old woman who was the next of kin in the remarkable *Bad News* performance for which Tarn Adams was fittingly the player.

Another example proved central to one of Ben Samuel's favorite emergent scenarios (the larger context of which I will introduce in the next chapter):

> This [character] came from a long line of restaurateurs himself—his great-great-uncle (for some number of greats) established a restaurant over a hundred years prior. This restaurant had been a staple of the town, and was passed down from father to son for many generations, until finally it was bequeathed to an owner who would never have any children of his own. When this owner chose to retire, he passed the restaurant on to his nephew, who had been a successful server at this restaurant all his life. However, less than a year later, the nephew managed to drive the restaurant into the ground; a century's worth of family history and town pride forever lost. After this, the nephew became a laughing stock of the town, ridiculed and reviled in equal measure. He became a recluse, hesitant to leave his apartment and face the vitriol of his community.[39]

In this case, the family restaurant was remarkable for a few reasons: it was in operation for over a century, and it was passed down from father to son several times in a row.[40] In each case, this is the result of the simple mechanics that I have outlined above. As I explained in Section 9.2.7, a business will shut down

[39] Personal communication, June 14, 2018.

[40] As I will explain in Section 10.6, in which I revisit this example at length, the restaurant in fact operated for 131 years.

on a given timestep if a probabilistic rule triggers. This means that a business that operates for a remarkably long time has simply been fortunate in terms of the random numbers that have been generated on its behalf. As for the notable succession of fathers and sons at the helm of the business, this is more dynamical, but still it emerges from simple mechanics: in this case, the structuring of the hiring process such that family members are preferred, which would have led each father, upon retiring, to score his son more highly than other prospective candidates. Finally, once the nephew took over, the business shut down after it had operated successfully for well over a century.

Our human minds leap to a causal explanation of this: the nephew was not cut out for the job, and the restaurant sunk like a ship; it was not his true inheritance, after all. But this is *apophenia*—the phenomenon whereby patterns of order are discerned when randomness is in fact at work [1203]—and so an overactive narrativization is at work. Really, the closure occurred because that probabilistic rule finally triggered. While most of the story was contingent on improbable series, the succumbing to probability in the long run is what proved poetic here. But, again, the real charge of this scenario is almost entirely due to a human proclivity to see narrative even where it is not.

The March of Progress

Another potent scenario produced by simple probabilistic mechanics concerned a character who owned a general store in a town for over fifty years, but was forced to close it after a new supermarket opened there. Ironically, and tragically, he spent his remaining days as a nighttime stocker at that very supermarket. Likewise, in one of our papers on *Bad News*, we recounted another scenario that also ended in tragical grocery stocking:

> [the player] explained [to the aunt of the next of kin] he was a historian

> chronicling the history of the town [...], hoping she would tell him more about her family, and ultimately the next of kin. The aunt obliged, telling familial history that reinforced the game's theme of loss: her father had been a town blacksmith for forty years of life, but as the march of progress advanced and demand for blacksmiths all but disappeared, her father lost his smithy and—after decades of being a skilled artisan—had to find work as a stocker at a grocery store, until he passed away [1090, pp. 117–118]

Both of these scenarios are essentially human narrativizations of the `demise` attribute of business definitions (see Section 9.1.3), which is what causes general stores and blacksmith shops to shut down around the same time that supermarkets begin to appear.

Design Pattern: Apophenia Hacking

In crafting a character simulation, it is helpful to anticipate that humans will unconsciously narrativize event patterns to identify emergent intrigue beyond what is actually generated by the system. In this way, apophenia may be targeted by a design pattern that I call *apophenia hacking*. In a keynote speech titled "The Power of Projection and Mass Hallucination: Practical AI in The Sims 2 and Beyond" [146], developer Matt Brown spoke about this phenomenon in the context of *The Sims 2* [793].[41] When it comes to *Bad News*, apophenia hacking undergirds a procedure of curation whereby Ben and I, as actor and wizard in the installation, process the raw emergent material of *Talk of the Town* towns to tell stories that are driven by an overactive narrativization. This is easy for us to do because we are humans, which means we cannot help but identify narrative patterns in the towns beyond what is modeled; I designed this simulation and still I cannot stop myself from noticing these patterns.

[41] Though the talk was not recorded, a summary by Mark Nelson is available online: `http://www.kmjn.org/notes/sims2_ai.html`.

However, in letting our apophenia run rampant, we constrain ourselves according to a narrativization motto: *augment but do not contradict.* That is, while our curation procedure is partly about *augmenting* the material of a town to tell stories that deal in richer phenomena than what is actually modeled in the underlying system, in doing so we do not contradict any of the explicitly modeled concerns. For example, romantic infatuation is modeled by the system, so we can know for sure whether a character is in love with someone else—a fact like that should never be contradicted by our overactive narrativization. In the next chapter, I will return to this kind of discussion, but the takeaway is that *Talk of the Town* appears to succeed at suggesting narrative patterns beyond those that can actually be produced by the system's explicit modeling.

Emergent Scenarios

To summarize, while it does not generate stories, *Talk of the Town* tends to produce an overabundance of narratively potent situations in the form of what I call *emergent scenarios.* These scenarios pertain to various simulation mechanics and social dynamics, several which I have outlined in this section: misanthropes, recluses, social status, the interpersonal circumplex, unrequited love, asymmetric friendship, internal conflict, love triangles (polygons), extramarital obsessions, business rivalries, town institutions, family feuds, family drama, and forbidden love. More powerfully, an emergent scenario may concern higher-order dynamics that richly recombine the concerns of multiple of these lower-order dynamics.

Unfortunately, the simulation of character knowledge phenomena provides little in the way of narrative intrigue, and this is due to its particular emphasis on character appearance. Again, this curious predilection results from the importance of character appearance in *Talk of the Town* (2015). While appearance is also important in *Bad News*, as I will explain in the next chapter, it is not important enough to vindicate the misappropriation of computation and authoring effort that could have been spent instead on having characters form knowledge about more interesting social concerns. Now, in *Hennepin*, I am exploring a more narratively potent subject of character knowledge: past events.

Nevertheless, in *Talk of the Town* the simple modeling of abstract social mechanics is rich enough to produce an array of intriguing dynamical phenomena, as I have shown in this section. In the following chapter, I will discuss how such narratively potent material is excavated for purposes of curation in the course of a *Bad News* performance. Thus, while this chapter has concerned the 'overgenerate' component of my tagline *overgenerate and curate*, the following chapter will make good on the second part.

Chapter 10

Case Study: Bad News

Adam Summerville and I had developed a prototype for a videogame about character knowledge phenomena—*Talk of the Town* (2015), as I explained in the previous chapter—but it was not hitting the mark. Following a critique session in Noah Wardrip-Fruin's class [1314], Adam emailed me to propose a revised game design. In the email, he noted that our initial asymmetric multiplayer design was all about character knowledge propagation, but our eventual simulation engine (the subject of the previous chapter) produced additional interesting phenomena that our original design did not showcase:

> I think an important part of [the class critique] was about the networks that people build up. We only surfaced that as knowledge propagation, but I think it's not just knowledge propagation. We need to get at friends and family, and enemies and rivals.[1]

Adam's proposed redesign, aimed at capturing all of the social intrigue modeled in the simulation engine, was a single-player realtime experience inspired by *The Legend of Zelda: Majora's Mask* [880]:

> As I've mulled on this, I think that something more akin to Majora's mask is the way to go. [...] In MM you meet up with a gang of kids

[1] Personal communication, March 13, 2015.

> early on [...] who give you a notebook. The notebook is important for 2 things: 1) As you learn about people's daily habits it gets put into the notebook. 2) As you learn people's goals/dreams/aspirations it also gets put into the notebook. The main part of MM (that isn't the traditional going into dungeons and killing things and getting items) is all about helping these people out. But to do that you first have to figure out what they do around town, where they'll be, who they interact with, and why. I think that something like that would be a much more interesting use of the simulation than our current game. [...] It could be the same kind of idea. You could be a Mary Poppins-esque character who rides into a town filled with unfulfilled people, and you have to follow them around, see what makes them tick and make them happier[2]

In the same email, Adam also emphasized the issues with our overwhelming menu system for conversational interaction. I am extremely interested in the prospect of freeform conversational interaction in videogames, in the style of *Façade* [776], and it seemed like the redesign could be an opportunity to explore this territory. At that time, Ian Horswill had recently published on *MKULTRA* [506, 509], a videogame prototype featuring a clever *autocompletion*-driven dialogue interface that suggests completions to partial free-text inputs, thereby guiding players toward utterances the system can actually understand.[3] In my response to Adam's proposal, which I was very excited about, I suggested that we could use this technology in our new design:

> As for the dialogue redesign—the more I think about it, the more I think we need to just allow free-text input. Ian Horswill is working on a game (underpinned by an open-source Unity framework) in which the player character is controlled by free-text input, but crucially he has a bidirectional parser that suggests autocompletions (of partially composed commands) that are commands that the system can actually

[2]Ibid.

[3]*MKULTRA* also features *belief injection*, whereby players reprogram character beliefs through free-text input, and *AI sonification* [508], by which character decision making (in an underlying Prolog engine) is cleverly sonified: "MKULTRA includes a sound synthesizer that is driven by a log of the activity of an NPC's problem solver, thereby allowing the player to 'hear the character think' in real time" [507, p. 224].

> understand. Unlike Facade, it won't accept a command that it can't understand. With the autocompletions serving as scaffolding in the (otherwise open) choice space, I think it's a really nice middleground between a straight-up parser and a menu with hundreds of choices. If we wanted to go this route, I could email him about getting started with integrating this into our project[4]

Adam was on board with the idea of using the *MKULTRA* technology, and so we emailed Ian Horswill, who was also excited about the prospect. Soon, this shift led to a major design decision: we would rewrite the *Talk of the Town* simulation engine using the *MKULTRA* framework. As I briefly mentioned in the last chapter, the integration of my Python simulation engine into the larger Unity project of our game proved to be a technical nightmare. Since Ian's technology was already fully integrated into a unified Unity project, it seemed advantageous to ditch the Python code and start from scratch using his code. Unfortunately, due to busy schedules all around, especially mine, our redesign and technical overhaul never got off the ground. It seemed that the project had fizzled out.[5]

Meanwhile, however, I had been gradually developing my own approach to conversational AI in games. Since arriving at UC Santa Cruz to work with the Expressive Intelligence Studio and Marilyn Walker's Natural Language and Dialogue Systems lab, my research agenda had primarily concerned the development of technologies that would enable freeform conversational interaction in games. Initially, this took the form of a project in which *Prom Week* [803] dialogue exchanges were decomposed into individual lines, which were annotated and then procedurally recombined to form new dialogue exchanges that could express previously inexpressible storyworld states [1052, 1053]. Through a gradual design evolution, the interface for dialogue annotation became a full-fledged authoring

[4] Personal communication, March 14, 2015.

[5] Ian is a friend of the lab whom I already knew outside the context of this missed connection. We have kept in touch since, and he is a member of the this dissertation's reading committee.

tool—the idea was that dialogue could be annotated *at authoring time* to enable intelligent recombination as the primary generative mechanism. Under the influence of *Tracery* [204, 205], a tool that had recently been created by my labmate Kate Compton, recombination was reformulated to work according to the operation of a generative grammar. As Adam and I were embarking on our game redesign, undergraduates Taylor Owen-Milner and Max Fisher produced a prototype of the tool, which I called *Expressionist* [1054] (see Section 12.2).

With a simulation engine already in place and an authoring tool for conversational AI developing, I had a wild idea that would incorporate the unique skill set of my labmate Ben Samuel. Here is an email that I sent to him in August 2015:

> I hope you've had a great summer so far! I have a crazy idea [...] and I'm writing to solicit your help with it. [...] I want to run a computationally assisted live-action prototype, where players interact with the game by: freely speaking to me (game master) to give commands about things like navigation, etc.; freely speaking to game characters to inquire about whatever they would want to inquire about. After the player asks a game character something, I would probe that character's mind to see what knowledge they have regarding the inquiry, relay it to a live actor playing the character, and then the live actor would improvise a response that expresses the particular bit of character knowledge that I relayed. [...] So, as you might expect, I'm writing to ask you if you'd like to be the live actor (you would play every character).[6]

Ben, who is now a professor at the University of New Orleans, is a unicorn: he is an AI researcher and a professional actor. Before I got to Santa Cruz, he had co-developed *Prom Week*, so he was no stranger to social simulation. As for acting, his performance in Hulu's first original scripted series, *Battleground* (2012) [1305], earned him praise from the *New York Times*:

> Mr. Samuel, who in addition to acting is working toward a doctorate in computer science at the University of California, Santa Cruz, combines

[6]Personal communication, August 13, 2015.

> elements of the "Harry Potter" Daniel Radcliffe and Matthew Gray Gubler's Dr. Reid on "Criminal Minds," and he's the best reason so far to watch "Battleground." [440, p. C2][7]

This idea that I proposed to Ben had actually emerged out of discussions with Jonathan Lessard, a professor at Concordia University in Montreal, whom I had met at the Foundations of Digital Games conference in the beginning of summer.[8] At the conference, Jonathan presented a poster titled "Design Rationale for Natural Language Game Conversations" [679], which reported on *A Tough Sell* (2014) and *SimProphet* (2015), two experimental videogames centered on freeform conversational interaction.[9] I was thrilled to encounter this work, and Jonathan was intrigued to hear about the development of both *Expressionist*, which he would eventually utilize in a later project [682], and *Talk of the Town*, which dovetailed with his own interest in character simulation (e.g., see [681]). We kept in touch over email after the conference to discuss the prospect of collaboration. Jonathan is a master of scope, which I am certainly not, and in one of our conversations about the *Talk of the Town* redesign—now conceived in large part as a testbed for my developing conversational technology—he suggested that I first explore the design space in a smaller project:

> For example: Find John Smith, he was last seen at Awesome Jeans [...] Other small puzzles could be: Someone died in apt. X, find out who lived there and then find his closest relatives (to notify them). A casting agency is looking for someone with X and Y facial features, find someone fitting that description.[10]

I loved Jonathan's ideas and decided to embark on a prototype that would follow the second premise: explore a *Talk of the Town* storyworld to find and notify

[7]Incidentally, *Prom Week* and *Battleground* debuted on the same day in 2012.

[8]Jonathan Lessard is also a member of the reading committee for this dissertation.

[9]Soon after, Jonathan released a third game in this vein, *SimHamlet* (2016), thereby culminating his *LabLabLab* trilogy [680]. I propose to call videogames like these—ones centered on conversational interaction—*talking simulators*.

[10]Personal communication, July 29, 2015.

the next of kin of an unidentified deceased character. With an early version of *Expressionist* now at my disposal, I was ready to start building a dialogue system for this videogame prototype.

Immediately, however, I encountered a design problem, which I explained in this way in my email to Ben:

> In setting out to build this game, I've come across a huge problem: I want dialogue to be the primary player interaction, but I want to be smart about writing content for the game, since the whole point is that it should be a small first exploration. More specifically, I want to write as little content as possible, but have that content pool closely align with the dialogue players would naturally want to say and hear in the game.[11]

As such, the idea that I had proposed to Ben—running a computationally assisted live-action prototype—was specifically aimed at gathering a corpus that could guide the authoring process:

> In the end, after audio-recording all player interactions, I'd have a corpus of natural/common character interactions in the domain of the gameplay experience, and I could then rely on this corpus during content creation.[12]

Needless to say, Ben was excited about this strange prospect and agreed to serve as actor. I reached out to Adam, as well, since the project was an offshoot of our work on *Talk of the Town* (2015), and he joined as a third codesigner (after writing back, "This is a crazy idea, but I kind of love it").[13] We called the experiment *Bad News*, and within a few days we began preparing for its installation as a demo at the upcoming Experimental AI in Games workshop [1060]. Improbably, within a year and a half of these humble beginnings, *Bad News* would be performed internationally at venues including IndieCade, Slamdance,

[11]Personal communication, August 13, 2015.
[12]Ibid.
[13]Personal communication, August 26, 2015.

and the San Francisco Museum of Modern Art, and featured in publications such as *Rolling Stone* [1359] and even on a television morning show [860].

A collaboration with Ben Samuel and Adam Summerville, *Bad News* is an installation-based immersive experience for one or two players.[14] Through its utilization of the *Talk of the Town* simulation engine, described in the previous chapter, the storyworld for each performance is procedurally generated prior to the start of gameplay. During gameplay, a player explores this storyworld and converses with its denizens in an effort to find and notify the next of kin of an unidentified deceased character. The installation is driven by a *wizard-of-Oz* configuration [235] that incorporates a human experience manager and human actor, who collaborate to curate the storyworld's emergent material into narrative artifacts that are mounted into the experience through the mechanism of conversational storytelling. While earlier publications have reported on several aspects of the project [1060, 1062, 1090, 1084], in this chapter I will provide a general overview before focusing discussion on its instantiation of curationist emergent narrative, the subject of this dissertation. Though I will also describe how the project has evolved over time, this chapter concerns its final instantiation.

10.1 Overview

Bad News combines wizard-of-Oz techniques [235] and live improvisational acting into an emotionally charged one-on-one experience whose story and setting is uniquely generated, for each performance, by the *Talk of the Town* simulation engine. Each performance takes place in the summer of 1979, at which time a resident in a computer-generated American small town has died alone at home, and a

[14] These names are listed alphabetically. Though we take various roles in the context of a particular performance—guide, actor, wizard—with regard to the creation of this work we each take the title 'codesigner'. As such, there is no meaning to any ordering of our names.

mortician's assistant—the player—is tasked with tracking down and notifying the next of kin. To do this, the player navigates the simulated town to interact with its residents, who are each played live by a professional actor. Throughout gameplay, an unseen wizard listens in remotely to manage the unfolding experience via live coding and discreet communication with the actor.

Antecedents and Related Pieces

Before describing *Bad News* at length, I would like to mention some of the few related pieces in the peculiar area of *computationally assisted live performance*. First, *Saga II* (1960) [859], the earliest known work of computational narrative, should probably be considered an example of this form. As I discussed at length in Section 4.1.2, in this collaboration between MIT and CBS-TV, seasoned Hollywood actors performed scripts generated by an early story generator. A more recent and better known example is *The Bus Station* (1993), a "live experiment" developed by Oz Project stalwarts Joe Bates and Peter Weyhrauch, in collaboration with playwright Margaret Kelso, as part of a deep exploration of drama management [575]. This piece placed interactors among improvisational actors in a tense scenario managed by a hidden director.

Earlier, around 1983, Brenda Laurel and Eric A. Hulteen explored similar territory in the context of the "media room" at the Atari Sunnyvale Research Lab. In a company memo, Hulteen describes the space:

> The media room itself is not so much a room in the traditional sense as a space or an environment. It has the capability to receive/understand/input all forms of human motor activity. Which is to say that it can sense any change that you can affect in the environment but that it can't read your thoughts. The room also has the capability to generate/output information to all of the human sense organs. It can provide things for you to see, hear, smell, feel, and taste. [517, n.p.]

In a follow-up memo that was written a few months later, Laurel and Hulteen propose an in-house experiment called *The First Fischell–Turing Production*, to be held in the media room:[15]

> The central goal [...] is to see how a first-person, interactive fantasy system implemented in the media room might work. We envision [...] a simulation of a single-user interaction with such a system. [...] The actual event will have the flavor of a planned improvisation, using operational technical elements where they exist and human improvisation for portions of the system that are still in the conceptual hacking stage. [657, p. 2]

Like *The Bus Station*, this experiment was intended to explore the prospect of doing drama management with a fully computational system:

> The project will utilize a structure which is based on the interactive story idea, with a playwriting expert system (*Play-Right*) [...] which orchestrates system-controlled events and characters so as to move that collectively generated story forward in a dramatically interesting way. [...] The operational elements of the system will include some interactive devices, video, graphics, and animation, sound, and other other environmental effects. [657, p. 2]

Laurel and Hulteen's articulation for the merits of this approach strikingly anticipates not only *The Bus Station*—which is not very surprising, considering that Laurel was an ancillary member of the Oz Project and its intellectual forebear—but also the impetus for *Bad News* that I recounted in the opening of this chapter:

> our purpose is to study the design of a system, major portions of which have yet to be implemented, by simulating its operation. The exercise will be improvisational in the sense that humans will enact roles that will, hopefully, later be automated [657, p. 2]

While I do not know for sure whether *The First Fischell–Turing Production* was ever performed, I have reason to believe it was. In a planning document that

[15]Here, 'Fischell' refers to Arthur "Artie" Fischell, an infamous fictional persona that Laurel and colleagues cultivated to the degree that Atari officials believed he was an actual employee of the company [340].

Oz figurehead Joe Bates drafted in preparation for the 1995 AAAI Spring Symposium on Interactive Story Systems—which I acquired from my advisor Michael Mateas, who photocopied it sduring his time as a graduate student on the Oz Project—a proposed agenda schedules video presentations on both Oz and Atari live experiments (presumably the ones I have mentioned):

> Interactive Drama Experiments from CMU, Atari [...] Joe [Bates] shows two 10 minute videos of Oz live drama experiments and video comments by actors, director, playwright, users after experiments end. [...] Brenda [Laurel] shows/presents the live drama experiments done and planned at Atari in the early 80s

In any event, a follow-up report by Laurel (also in 1983) laid out more extensive details on the planned implementation [653]. Eventually, her ideas about interactive fantasy would solidify in her 1986 PhD thesis [654] and the seminal 1991 monograph *Computers as Theatre* [656].

Later obscure projects in the area of computationally assisted performance include *Pleasure Island* (1999), a networked "online drama" pitting interactors against improvisational actors [178], and *Terra Incognita* (2002), a live performance incorporating robot actors [1130, 1131]. *AR Façade* (2006) was a gallery installation that mounted *Façade* [775] in an *augmented reality* [58] environment, with human operators intervening live to guide its reactive-planning ecosystem [272, 273, 271]. *Party Quirks*, by Brian Magerko and collaborators, is an agent-based mixed-initiative framework for playing a popular improv game [725, 903]; generally, Magerko and his collaborators have been active in this area for the last decade [726, 534, 494, 533]. In 2012, startup Alternate Reality produced a networked game that placed improvisational actors in precarious physical situations, which players were tasked with rectifying with the help of other in-game characters played by humans [577]. More recently, Rob Wittig and Mark C. Marino have developed the *netprov* form [740, 152], which they describe like this:

> A typical netprov, if such can be described, involves a networked of platform (such as Twitter, a blog, or Facebook), a set of collaborators, a narrative premise, and a set of constraints. For the most part, netprovs have time limits, although that is not a necessary requirement. Fundamental to netprov is creative play on contemporary platforms. [1346, p. 4]

While generally we found out about these projects later on, one antecedent directly inspired the design of *Bad News*: Dietrich "Squinky" Squinkifer's *Coffee: A Misunderstanding* [1189], which is a computationally assisted interactive play that Squinky has described in this way:

> Each character in "Coffee" is portrayed by two audience volunteers. One, the "driver", is given sets of branching dialogue options at key points in the story, which are selected using a menu displayed on a mobile device. The other volunteer plays the "puppet", who recites the dialogue selected by the driver, the lines for which are also displayed on a mobile device. Meanwhile, the other audience members are able to see each character's dialogue options via a background projection, and are encouraged to heckle the performers and attempt to sway the drivers in their choices. [1189, p. 1]

Squinky, a former UC Santa Cruz graduate student, ran a performance of this piece as a class project in Noah Wardrip-Fruin's *Playable Media* class [1313], which I thoroughly enjoyed as an audience member. While Ben and Adam were not in the class, they were aware of the project, and it was almost certainly in the back of my head as the initial ideas about *Bad News*'s peculiar configuration began to take shape. Squinky has since produced additional games in this tradition with collaborator Jess Marcotte—together, they form the collective Handsome Foxes in Vests. Here is their description of *Most Sincere Greetings, Esteemed One* (2016):

> Each pair of players is asked to perform a set of procedurally-generated instructions for greeting one another, then taken to a page where they receive procedurally-generated feedback on how well (or poorly) they executed the greeting. [1185, n.p.].

Caressing a cybernetic plant in *rustle your leaves to me softly* (2017) triggers a sensual procedural soundscape [1186, 737]. Finally, *The Truly Terrific Traveling Troubleshooter* (2017) is a "radically soft physical/digital hybrid roleplaying game about emotional labour and otherness, which fits entirely inside of a carry-on suitcase" [1187, n.p.], that being the titular Troubleshooter's toolkit [1188, 738].

I am aware of two projects in this growing area that have actually been inspired by *Bad News*. Ian Horswill's *Dear Leader's Happy Story Time* (2016) [511] leverages a story generator that operationalizes media tropes to produce kitschy plot outlines that are meant to be acted out by a group of friends as a party game. As Horswill explains, the project very intentionally leans into the clunky and mechanical aesthetics that characterize the outputs of story generation (a notion that I discussed in Section 3.1.4):

> The approach taken in *Dear Leader* is to salvage bad stories through playful, ironic performance; calling attention to their artificiality and theatricality; and surfacing the mechanistic character of their generation. [511, p. 40]

Horswill's project pushes further into the territory he explored in the earlier *Fiascomatic* (2015) [510], a scenario generater that produces storyworld representations that are meant to be used for Jason Morningstar's tabletop storytelling game *Fiasco* [857]. In each of Horswill's projects, computational systems produce material that human interactors are meant to act out or perform in some way.

The other project in this area that has been inspired by *Bad News* is *Hot Bread* (2017), a project by researchers Rossana Damiano, Vincenzo Lombardo, and Antonio Pizzo, and their students at the University of Turin [238]. To create this piece, Lombardo used *Talk of the Town* to generate a storyworld whose emergent material was explored over the course of multiple weeks in a class on interactive storytelling.[16] Eventually, the students homed in on a romance between two char-

[16] At this time, town seeding was not working properly in the simulation engine, which meant

acters who worked at rival bakeries in the town, and they used this material to build *Hot Bread*, a work of curationist emergent narrative that hooks into the researchers' *DoPPioGioco* framework for computationally assisted live performance that uses computer vision and emotional modeling [237]. The following year, I visited Italy to collaborate with this group on what became *Cattive Notizie*, a variant of *Bad News* that I describe in Section 10.2.5.

Finally, I will note that there is also growing scholarly interest in the scholarly area of computationally assisted live performance, as a number of recent papers attest [748, 1091, 1206, 57].

10.2 Installation and Experience

I will now provide a fairly extensive description of *Bad News*, but again my primary goal in this chapter is to provide a case study that considers the experience with regard to my framework for curationist emergent narrative.[17] For other reports on the project, see our earlier conference papers [1062, 1090] or Ben Samuel's dissertation [1084]. Additionally, an earlier workshop abstract provides intellectual context on our initial conception of the project as a live-action prototype that would guide content creation for a digital game [1060].

10.2.1 Premise

It is the summer of 1979 and a resident of an American small town has died alone at home. The death has come to the attention of the office of the *county*

that Lombardo had to maintain the storyworld on his laptop for the duration of the experiment. Unfortunately, toward the end of the scheduled period for story sifting, he forgot about this and restarted his computer, and as a result the world disappeared forever.

[17] The overview provided in this section adapts the prose of a design document for the project titled "Bad News: Behind the Curtain" [1048], which we have published on our project website.

mortician, though the event is not known in the town itself. The cause of death is unknown, but foul play is not suspected.[18] In the town's jurisdiction, it is the responsibility of the county mortician's office to notify the next of kin of any death. Unfortunately, the county mortician is occupied by another matter in the county, which means his newly hired assistant (the player) is tasked with handling this particular case. Given a description of the body and the address at which it was discovered, the mortician's assistant must use this information to: 1) determine the identity of the deceased person, 2) determine the identity of the next of kin (closest familial relation in the town), and 3) track down and notify the next of kin of the death.[19] Again, there are no indications of foul play, and indeed it is not the player's responsibility to determine the cause of death; she must simply find and notify the next of kin.

10.2.2 Personnel

There are five distinct roles that are central to a *Bad News* performance:

- **Player.** The *player* is the person who is playing *Bad News* for a given performance. In some cases, we allow for multiple players, but generally it is intended as an intimate immersive experience for one player.

- **Guide.** The *guide* is a team member who introduces the player to the narrative premise and our gameplay interactions in a brief session preceding a performance. Additionally, in an exhibition setting, the guide serves

[18] *Bad News* is very intentionally not a murder mystery, but rather a meditation on the splendor of every life. While fortunately the death of a loved one is not an everyday matter in the context of our individual lives, it is an everyday matter in this world. This is what the piece explores.

[19] More specifically, the next of kin is the closest living familial relation in the town, where closeness is defined according to the following hierarchy: spouse, parent, child, sibling, extended family. In many cases, there may be multiple next of kin, meaning multiple characters who are equally closely related to the deceased character. For example, if a character has several living siblings in town (and no spouse, parent, or child there), then each of the siblings is a next of kin. In such cases, notifying any single one of them will suffice.

Figure 10.1: *Bad News* guide Adam Summerville addresses a crowd gathered around an installation at the San Francisco Museum of Modern Art. The guide is tasked with introducing players to the experience and engaging with passersby in exhibition settings, such as this one.

as a public-facing team member who is free to discuss the project with passersby, as Figure 10.1 shows. This role is fulfilled by codesigner Adam Summerville.[20]

- **Actor.** The *actor* is a team member who acts out every character that appears in a given performance. Additionally, the actor plays the mortician in an opening sequence (more on that below). This role is fulfilled by codesigner Ben Samuel.

[20]In a few installations, Tyler Brothers has filled in for Adam. Alternative names for his role that Adam has nominated include 'carnival barker' and 'den mother', the latter capturing his proclivity for making sure Ben and I are fed and hydrated in what can be eight or ten hour sessions of back-to-back performances.

- **Wizard.** The *wizard* is a team member who listens in on gameplay via a microphone to manage the experience in real time. His primary duties are executing player commands and discreet communication with the actor. As I will explain below, much of the wizard–actor communication concerns the delivery of narratively potent material sifted out from the simulation—as such, a critical activity of the wizard is *story sifting*. In exhibition contexts, the wizard may perform his duties in a public-facing area, as a kind of performative *livecoding* [200]. I fulfill the wizard role in *Bad News*.

- **Simulation.** The *simulation* is an instance of the *Talk of the Town* framework, which was the subject of the previous chapter. This contains the unique procedurally generated town that will serve as the storyworld for the performance. The town is generated just prior to a performance and then kept in memory, so that the wizard may explore it by constructing the appropriate Python queries. Additionally, aspects of the storyworld are expressed in the actor and player interfaces, as I discuss in the next section.

10.2.3 Installation

Bad News is an installation-based experience that is driven by a *wizard-of-Oz* [235] configuration that integrates physical and technological elements. Figure 10.2 shows a performance at an installation that was built for the Slamdance DIG showcase, held in Los Angeles in December 2016. Our installations have varied from exhibition to exhibition, but there are several critical components that must be in place in order for the piece to be performed:

- **Gameplay area.** A *gameplay area* houses a table, two chairs (on opposite sides), and our custom *model theatre*, which is the centerpiece of the installation. On the actor's side of the table is a computer, hidden from the

Figure 10.2: A *Bad News* performance at the Slamdance DIG showcase, held in Los Angeles in December 2016. The *wizard* (left) listens in and livecodes, while on the other side of the facade the *actor* (far end) has drawn the curtain of the model theatre to converse with a pair of players. A discreet screen positioned behind the fixed lower half of the curtain displays an *actor interface*, and the players take notes and view a tablet that displays a *player interface* with information about their current location. To move about the town and take actions beyond conversation, a player simply speaks what she would like to do aloud—the wizard, who listens in via microphone, then livecodes to change the storyworld accordingly, with the effects being expressed in both the actor and player interfaces.

player's view by a fixed lower curtain on the model theatre; this computer displays the *actor interface*, which I will discuss below. On the player's side of the table is a notebook and pen (for taking notes), as well as a tablet computer; the latter displays the *player interface*, which I will also explain shortly. Additionally, a microphone is placed on the table—the actor's computer captures live audio that is conveyed to the wizard's computer over a local area network.

- **Wizard command center.** In a separate area, the *wizard command center* is set up. For some performances, this has been a separate room, or even a separate city: using the internet (instead of a local network), we carried out multiple performances at the Slamdance DIG showcase in which I managed Los Angeles gameplay from my living room in Santa Cruz.[21] In other performances, this area is public-facing or even on stage before an audience, turning the wizard's activities into a kind of metaperformance based on performative livecoding (and live commentary from both the wizard and the guide); I will discuss the latter configuration at more length below. In any event, this area houses the wizard's computer, and potentially an array of additional screens to display the player and actor interfaces that I manage.

- **Partition.** A *partition* separates the gameplay area from public areas. If the gameplay area does not have a whole room to itself, we install some kind of partition that may separate it from a public area with attendee traffic. We do this to maintain the intimacy of the experience—a core design goal—and to ensure that the player does not feel like she is performing for an audience. In some installations, such as the one pictured in Figure 10.2, this partition has worked to both isolate the gameplay area and to separate the gameplay and wizard areas. (Though it is not visible in the image, a curtain from the right edge of the facade to the wall behind the camera could be

[21] In fact, I piped the audio through a sound system, and my wife and I enjoyed each performance in the style of a radio play.

closed to cordon off the gameplay area.) Additionally, it can be used as a public-facing facade that lends visual intrigue to the installation.

10.2.4 Experience

The *Bad News* experience proceeds in three distinct phases: lead-in, gameplay, post-mortem. In this section, I will provide an overview of what occurs (both in gameplay and behind the scenes) during each of these phases. While each performance tends to total about an hour, core gameplay (in which the player is exploring the town to speak with residents) generally lasts around forty minutes.

Lead-In

The *lead-in* phase begins when a player arrives at the installation for her scheduled performance. At this time, the guide takes her aside to explain what will happen, while the wizard and actor begin to prepare. This all typically takes about five minutes. Specifically, this phase entails the following procedures:

- **Guide preliminaries.** The guide meets with the player to introduce her to the narrative premise and the game's mechanics. The latter are also captured in a reference sheet, which he provides to her; this sheet contains a list of actions that the player can take during gameplay by simply speaking them aloud (the wizard will hear the command and livecode to execute it).

- **Town generation.** The wizard starts up the *Talk of the Town* simulation engine and uses it to generate a storyworld (by the procedures explained at depth in the previous chapter). Once a specific date in August of 1979 has been reached, the procedure generates the *narrative premise* by automatically selecting a character in the town who will be the deceased character

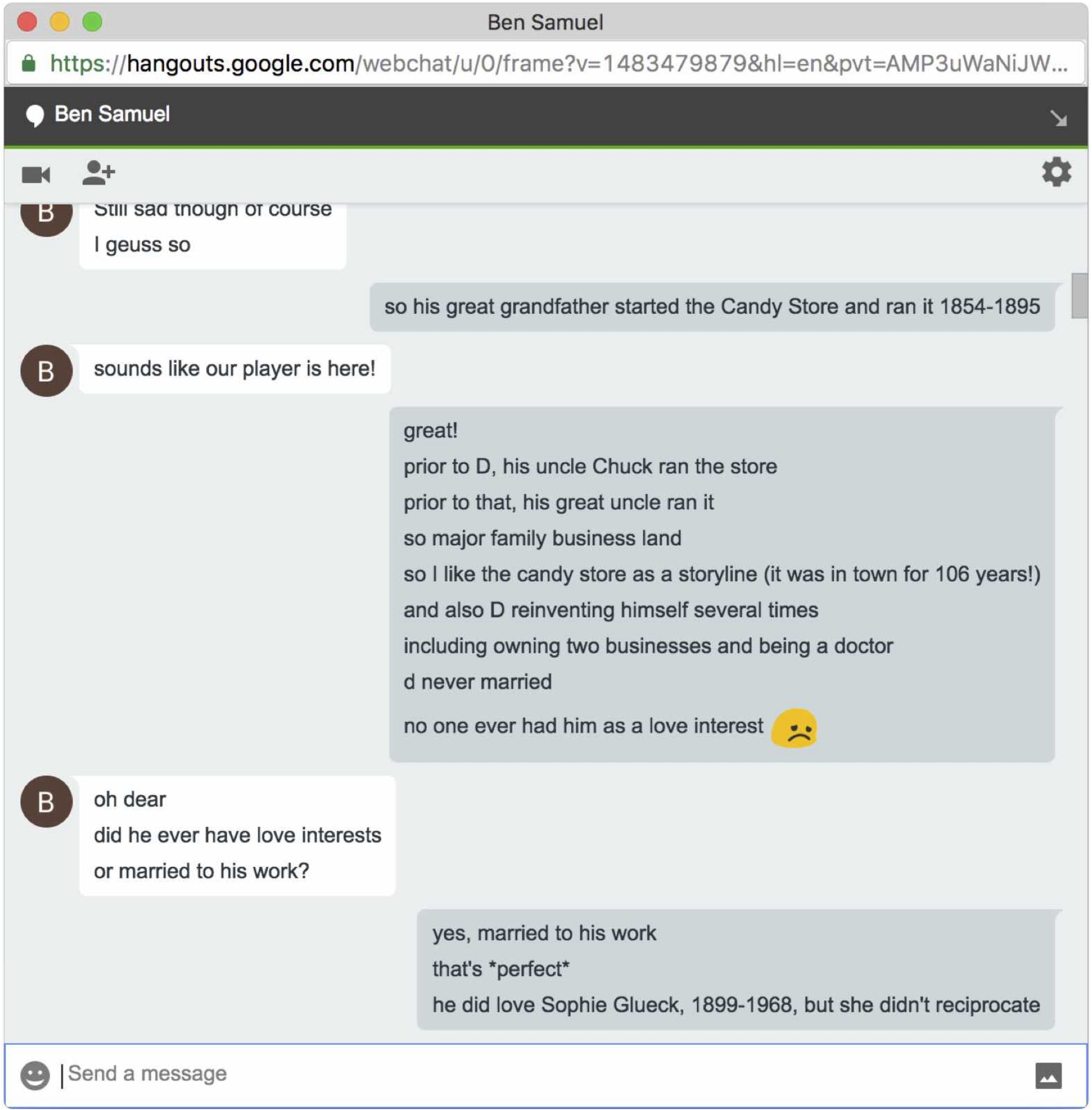


Figure 10.3: A web chat between the *Bad News* wizard (me) and actor (Ben Samuel). Prior to a performance, as well as throughout, the wizard explores the storyworld's generated material in search of nuggets of dramatic intrigue, which he feeds to the actor via the chat (the actor's computer screen is hidden from the player). As part of this interaction, the two settle on a set of *emergent scenarios*—ones that happened to have obtained in the town's simulated history—that the actor will attempt to reveal to the player naturalistically, through conversational storytelling. This is curationism at work: the wizard is *story sifter* and the actor is *narrativizer*. In the shorthand of our communication scheme, 'D' refers to the deceased character for the upcoming playthrough.

in this performance, and then having that character die at that time. To be precise, this selection (and death) occurs at the level of the *Bad News* code layer, which is driven by, but decoupled from, the *Talk of the Town* engine. Here, instead of random selection, the system employs some light reasoning, as it attempts to target a character whom most of the town knows, and makes sure that the selected character is home alone and has at least one relative in the town.[22] Additionally, this set-up procedure identifies the *next of kin* of the deceased, given the definition of it that is operational in *Bad News*. The entire procedure of creating a town and preparing the narrative premise typically takes between two to three minutes. As I noted in the last chapter, it is not feasible to generate many towns at the beginning of an exhibition day, since each takes up several gigabytes of memory.

- **Initial story sifting.** Once the town has been generated, the wizard (me) begins to explore the generated simulation material in search of nuggets of dramatic intrigue—this, of course, is the curationist task of *story sifting* (see Section 5.3). In doing this initial story sifting, I isolate a set of *emergent scenarios* of the kind discussed in Section 9.3. Again, these are narratively potent situations that happened to have emerged in the particular storyworld in which gameplay is about to take place. In the case of *Bad News* story sifting, my aim is to uncover scenarios that involve the deceased character in some way, since that person's life is the structural scaffold that will undergird the performance. Just as the randomly selected character Yvette anchored an exploration of a generated town at the Workshop on Tutorials in Intelligent Narrative Technology, as Figure 9.6 illustrates, so too does the deceased character in this case. As such, in my initial story sifting I

[22] In early performances, the selection was random, which led to a case of a town hermit whom no one actually knew and whose next of kin was an estranged cousin.

may encounter a love triangle involving the deceased character, or a secret affair or family drama, or any of the types of scenarios outlined in Section 9.3. Typically, during this period I identify between three and five scenarios, which I deliver to the actor—via our chat, explained next—who then adopts experience-level goals such that these emergent scenarios become conversational stories that he delivers to the player. That is, the actor will attempt to naturalistically reveal these scenarios to the player through conversation during gameplay. Beyond gathering a set of potent scenarios, I will more generally explore the deceased's person life history—family history, work history, social networks, friends, enemies, romantic history, love interests, characters in love with them, and so forth—to get a better sense of who this person was. Again, such excavated material may be delivered to the actor.

- **Actor priming.** In the period preceding gameplay, the actor is already stationed behind the model theatre, which has its curtains drawn. As I have noted, the wizard and actor begin communicating during this period via a live web chat (see Figure 10.3). During the process of initial story sifting, the wizard feeds the actor information on the newly created storyworld, as the two prepare for the next performance. This is how the actor finds out about the deceased character, next of kin, town, and anything else that is of importance to the playthrough. Critically, this is where the wizard and actor collaborate to isolate a set of emergent scenarios that the actor will work to naturalistically reveal through conversational storytelling.

- **Entering the gameplay area.** Once the player is ready, the guide escorts her into the gameplay area, where she takes a seat on her side of the table. Here, the guide hands her a notebook, pen, and the tablet computer—as he explains, on the latter a *player interface* will display information about her location as she explores the town during gameplay, as Figure 10.4 illustrates. To start, the interface shows an initial prompt that reiterates the guide's explanation about how to take action and asks for the player to speak aloud when she is ready to begin. At any point, the wizard may display arbitrary text onto the screen, either as a means of communicating directly to the player or of describing some aspect of the storyworld that is not typically rendered in the interface text.[23]

Gameplay

At this point, the player is sitting on her side of the curtain and the actor sits across from her, though she has not seen him yet due to the curtain being closed. The guide has by now left the gameplay area and is likely interacting with exhibition passersby, and the wizard remains out of sight at his command center, where he awaits the player's indication that she is ready to start. From here, the experience proceeds as follows:[24]

- **The death scene.** Once the player says that she is ready to begin, the wizard updates her interface to show a new prompt indicating that she is alone at the *death scene*. Critically, this prompt provides information about

[23] A favorite usage of this mechanism transpired in the course of researcher Jessica Hammer's playthrough at the 2016 ACM CHI conference. Upon observing a potential next of kin at a deli, Jessica returned to the home of the deceased character to search for photographs that would show him with someone matching the appearance of the prospective next of kin. While the simulation does not model home interiors—and certainly not artifacts like photographs—this clever move seemed worthy of reward, and so I extemporaneously composed a textual description of a photograph, hanging on the wall, that depicted the deceased character arm in arm with a person having the appearance under question (since Jessica's suspicions about the next of kin were correct). This was risky, since it required salting my hastily composed message with function calls (to insert a physical description of the next of kin), and any typo in this code would have caused the player interface to screw up at an intense moment in the experience. Remarkably, what I typed computed, and the moment worked beautifully. This was perhaps my proudest moment as wizard.

[24] For an outline of an example playthrough, see our conference paper on the project [1090, pp. 117-118] or Ben Samuel's dissertation [1090, pp. 375–378].

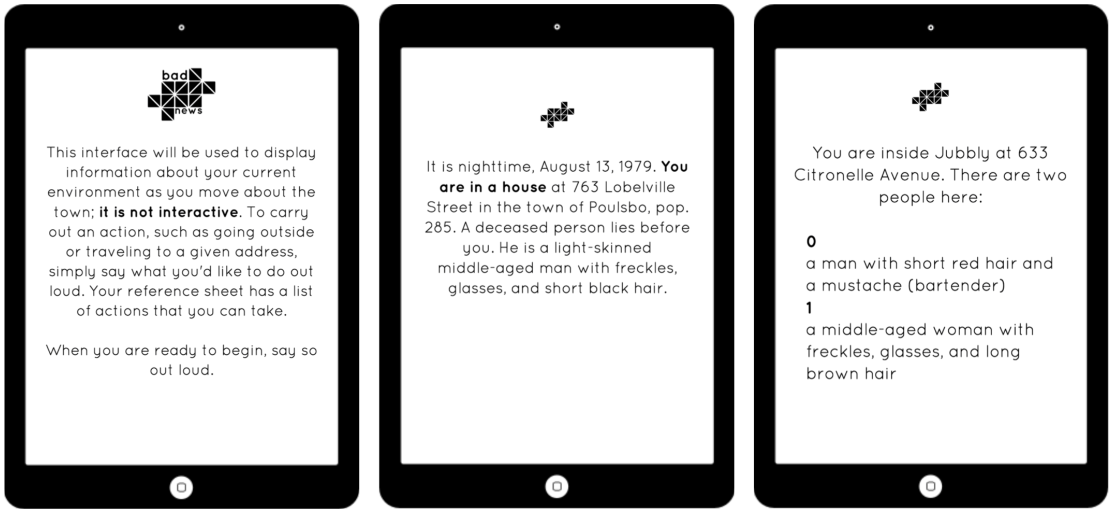


Figure 10.4: Example prompts displayed on the *player interface* used in *Bad News*. Throughout gameplay, a tablet screen is used to deliver noninteractive prompts to the player, primarily in the form of descriptions of her surroundings. Shown here are the initial prompt (left), a death-scene prompt (middle), and a typical location description (right). When the player sits down at the model theatre to commence gameplay, the initial prompt is displayed. Once she indicates that she is ready to start, the death scene begins, and the corresponding prompt is generated for display on the player interface. From this point on, the player says what she would like to do out loud—for example, "move one block north"—and the interface updates to describe her new location, if applicable. In each case, it is the responsibility of the wizard to listen for player commands and maintain the player interface through livecoding (updates are executed through the use of a local area network or the internet). The player interface is meant to be spare and minimal—it should not detract from embodied conversation with the actor, which is the core interaction in *Bad News*.

the deceased person's appearance and home address—the player will need to use this initial information to determine the identity of this person. Figure 10.4 shows an example prompt.

- **The mortician scene.** A few moments after the death-scene prompt is first displayed, the actor is cued by the wizard to begin the *mortician scene*. At this point, he opens the curtain and reveals himself as the county mortician (recall that the player's diegetic role is assistant to the county mortician).

The mortician apologizes for arriving late and informs his assistant that he must leave to handle a matter elsewhere in the county, which means that the player must track down and inform the next of kin on her own. Gently, and in detail, he outlines what this will entail, and encourages her to take notes (see Figure 10.10 for an example of player notes taken during gameplay). To respect the privacy of the family of the deceased, the mortician suggests that he and the player collaborate to come up with a *cover story* for her, since it would be tactless to openly parade around the town as a mortician's assistant.[25] Additionally, the mortician asks the player to participate in a brief imaginary conversation with a town resident, as practice. For this, he draws the curtain of the model theatre, pretends to be a person hearing a knock on his door, and then opens the curtain to engage in conversation with the player. Finally, the mortician fields any last questions before closing the curtain to exit the scene; from here, core gameplay begins. The mortician sequence is the only scene in *Bad News* that has a scripted structure, and it is intended as a sort of diegetic tutorial that onboards the player for our peculiar mode of gameplay (and particularly the kind of role-playing and improvisation that it entails). It can take anywhere between five and fifteen minutes, depending on the initial comfort level of the player—we extend it until the player appears to be ready to proceed—or other various concerns.[26]

[25]We've found that having a cover story—on top of the narrative framing of being the mortician's assistant—serves as a sort of scaffolding that helps players who are uncomfortable with role-playing to improvise more readily. Example cover stories include a county historian, genealogist, private investigator, lottery representative, and more.

[26]In a recent personal communication (June 14, 2018) in which I asked about his favorite *Bad News* experiences, Ben Samuel recalled a notable mortician scene: "Another memorable playthrough involved Tarn Adams of Dwarf Fortress fame, whose playthrough was perhaps most notable for having a thirty minute long mortician scene, when the typical duration averages around five minutes. In addition to the usual conversation pieces centered around introducing the player to the nature of speaking with an improviser and helping them to develop their cover story, this mortician sequence included philosophical musings on the nature of life and death, and the unique role our profession has in bridging these two disparate states of being."

- **Gameplay in three acts.** I loosely conceive of *Bad News*'s core gameplay sequence (in which the player explores the town to speak with its residents) as having an experience arc with a three-act structure:

  - **Act I: Learning the name.** Players typically spend the first five or ten minutes of core gameplay working to learn the deceased person's name.[27] Here, there are a number of strategies that players have taken: do a reverse look-up in the city residential directory, using the person's address;[28] inquire with a neighbor (using the cover story); head to a hot spot in town, such as a bar; and many more. To take an action, such as viewing the city directory or navigating to a new location, the player simply says what she'd like to do out loud—the wizard is always listening in, and it is his job to promptly update the player interface after each command. Fig. 10.4 shows the player interface after a player has navigated to a bar. Wherever she goes, the player is free to engage nearby characters in conversation; again, this is done by speaking aloud. Whenever she does this, the *actor interface*, shown in Figure 10.5, updates to show information about the character with whom she is starting a conversation. At this point, the actor scans the interface to learn that character's personality, history, and life data, and then spends a few moments getting into character (consciously adjusting his stance, mannerisms, intonation, and so forth accordingly).

[27]Note that the wizard and actor can coordinate to provide this information more easily if a player is having a hard time finding the name.

[28]This strategy was first employed by Ben Spalding, who at the time was affiliated with the Expressive Intelligence Studio as a student in UC Santa Cruz's Digital Arts and New Media MFA program. Ben was our most frequent early playtester and provided invaluable insights that guided an initial series of design iterations. This particular strategy was innovated as part of Ben's individual exploration of what *high-level play* [1265] would mean in the unlikely context of *Bad News*. From our standpoint, his *metagaming* [127] led to the identification and development of a series of levers with which we could exert control over the experience.

Once prepared, the actor opens the curtain to play that character live. Throughout a given portrayal, the actor must constrain himself so as not to contradict the character's personality, beliefs, or life history—this requires glancing at the actor interface and the web chat with the wizard as needed, which Ben does with expert discreetness.[29] The actor is allowed, however, to *augment* the simulational material with additional detail.[30] When a conversation ends, the actor closes the curtain of the model theatre and the wizard waits for the next player command. In his own dissertation, Ben writes at length about the experience of acting in *Bad News* [1084, pp. 369–375].

– **Act II: Tracking down the next of kin.** Once the player has determined the deceased person's name, the remaining work is to determine who the next of kin is and where that person is currently located. This is the phase in which the core experience of *Bad News* begins to open up. During this period, the player must ask around the town about the deceased character and the life that person lived—as such, it is primarily during these conversations that the actor seeks to reveal the emergent scenarios that were isolated as narratively potent during the wizard's initial story sifting. Here, the tone of the piece tends to become more intimate, as the townspeople begin to open up more to the player (and the player begins to open up more in her own improvisation). Through these conversations, the player not only discovers the

[29] In fact, several players have been surprised to discover, after gameplay, that Ben even had a hidden screen on his side of the theatre.

[30] For instance, in one generated town, the player encountered a character drinking alone at a bar—a house painter in his mid-twenties with few friends in the town. Though the simulation does not model character ambitions, the actor augmented the character's base simulational material to improvise a scene about him dreaming of leaving his small town to become an artist in New York City. When it comes to this kind of narrativization, our motto is *augment but do not contradict*, as I have already noted. I will discuss this example at more depth below.

identity of the next of kin, but more critically she learns about the life that the deceased person lived. Often times, information uncovered during this phase will raise the emotional stakes of the impending notification.[31] This 'act' typically lasts between twenty and thirty minutes, and it culminates once the player has determined the location of the next kin and traveled to that spot.

– **Act III: The notification scene.** This scene may occur at either a home or a public place—it all depends on where the next of kin happens to be located in the simulation. Often, a player takes a few moments to collect herself before engaging in conversation with the next of kin. Preceding this scene, Act II can vary considerably in tone, depending on the player's improvisation and the particular characters and storylines that she encounters; it often features comedic moments. Act III, however, is always emotionally intense. Players frequently ask the next of kin to sit down (even though the actor is already sitting down in physical reality), or request to take the interaction to somewhere more private in the storyworld. In several cases, the actor and/or player have begun to weep.[32] Once the notification has been processed by the next of kin—the culmination of which depends on Ben's performance—the actor closes the curtain and the player interface updates to express that the player has successfully completed her task: "You have successfully

[31]In one playthrough, the next of a kin was an older women who twice prior had married an older man who died of natural causes. After the death of her second husband, she instead married a young man in his twenties—tragically, this man was the deceased character for the playthrough. In mentally preparing for the notification scene, the player spoke aloud to himself about the difficult prospect of delivering bad news to a woman who had already heard it on two prior occasions.

[32]In one of the design team's favorite moments, Noah Wardrip-Fruin reached his hand across the model theatre as a gesture of consolation, which the next of kin accepted in a tender scene.

notified the next of kin. You have delivered the bad news."[33]

Post-Mortem

While core gameplay ends when the curtain closes for the final time, the experience is not yet over. As part of our *contract of care*—a practice in immersive-experience design that entails the development of scaffolding to ensure greater experiencer comfort [721, p. 150][1206, 227]—we carry out a gentle off-boarding process through which the player may be eased back into the real world after her emotionally intense visit to a *Talk of the Town* town.[34] Additionally, in this closing period, the experience converges back onto the simulation procedure that enabled it in the first place: in an *epilogue* scene we use *Talk of the Town* to simulate ahead in time to see what became of all the performance's principal characters. What follows are the final phases of the *Bad News* experience:

- **Decompression.** After the curtain has closed for the final time, we allow the player a few moments to collect herself and properly exit the storyworld. This means waiting to enter the gameplay area until the player's body language expresses that she is ready to exit the storyworld and speak to the guide, wizard, and actor. Typically the actor himself will likewise require a few moments to collect himself and return to the role of Ben Samuel.

[33] I should note that it is entirely possible for a player to notify the wrong person. According to the rules of the experience, the game ends when the player tells *anyone* about the death—that is, any divulging is treated as the notification (because otherwise the family's privacy is not respected). While the next of kin has been successfully notified in nearly every single playthrough, there have been a few cases of the wrong person being informed. In an unfortunate testament to *Talk of the Town*'s emergent serendipity, one player duo notified the next of kin of a living character who improbably shared all of the peculiar physical attributes of the deceased, including a noticeable tattoo, birthmark, and scar. A stranger incident concerned a player who successfully tracked down the next of kin but then elected to leave that scene to instead notify the janitor at the deceased character's apartment complex.

[34] For more on the *Bad News* contract of care, see Ben Samuel's dissertation [1084, pp. 366–368].

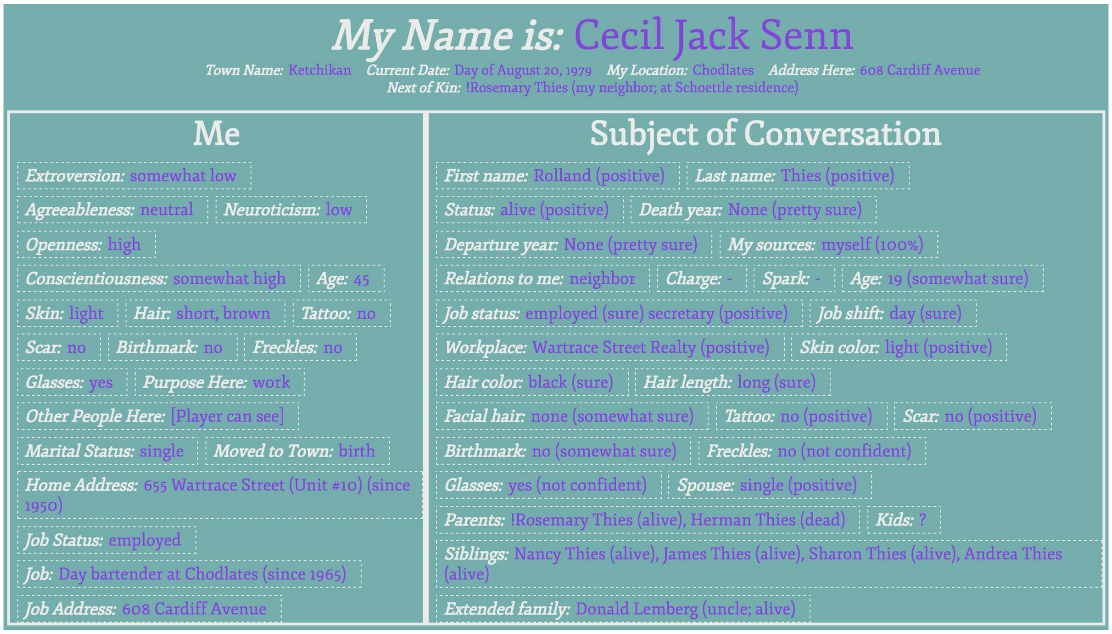


Figure 10.5: A screenshot from the *actor interface* utilized in *Bad News*. Throughout gameplay, this interface dynamically updates to display critical information in support of Ben Samuel's embodied improvisational performances of *Talk of the Town* characters. When the player decides to initiate a conversation with a character, the interface updates to display basic information about that person. Before drawing the curtain back to begin the conversation, the actor spends a few moments studying this information to determine how he will portray the character. The interface expresses (on the left side) the character's personality, appearance, employment details, and life history. As the conversation shifts between different subjects of conversation (other characters that the player and actor are discussing)—for example, by the player asking, "So who is this realtor?"—the right side of the interface updates to show the beliefs that Ben's character has about the new subject (including confidence levels). These updates are triggered by the wizard's livecoding, and all of the actual character beliefs will have been accumulated according to the procedures described in Section 9.2.12. In this example, the player and a bartender named Cecil Senn are discussing a realtor named Rolland Thies.

- **Post-mortem.** Once it feels appropriate, the guide and wizard enter the gameplay area and thank the player for playing; the actor emerges at this time to do the same. From here, we ask the player for her thoughts, and broadly discuss the experience, as a kind of informal postmortem.

- **Epilogue.** Finally, we escort the player to the wizard command center for a brief *epilogue* scene. Just as *Talk of the Town* can simulate the history of a town up until the summer of 1979, it may also simulate the *future* of a town from that point on. This simply means running the simulation procedure of Section 9.2 again, but this time with a later termination date.[35] Specifically, I simulate up to the summer of 2009—thirty years after the time in which the performance took place—and then explore the emergent material to determine what happened to the principal characters and locales that the player encountered during her visit to the town. Often, many of the principal characters will have themselves died—the simulation will say on which date—but updates can be positive as well (such as a deceased character's business still thriving, or young children in 1979 becoming prominent town figures by 2009). Sometimes, the epilogue is poignant: in one playthrough, the deceased character's brother was the next of kin, and also the town realtor—here, the simulation revealed that after receiving the death notification he had to carry out the sale of his deceased sister's now-vacant home. Unfortunately, however, nothing that the player does during gameplay alters the course of the future of the town. This is because gameplay occurs at a much higher level of detail than *Talk of the Town*'s simulation, which makes it infeasible to impute what the player has done back into the actual storyworld. Nonetheless, however, apophenia is always at work in human interpretation (see Section 9.3), which means players are likely to mistakenly discern ways in which the storyworld was changed by their actions. I will discuss a powerful example of this kind of apophenia in Section 10.4.

- **Goodbyes.** Once the epilogue concludes, the player's *Bad News* experience

[35]It was not until after fifty or so performances of *Bad News* that I realized the feasibility of doing an epilogue.

is over. No one had ever been to the town that she visited, and no one will go there again. I close the console window and it disappears forever.

10.2.5 Variants

While our installation and core gameplay experience are not variable, in a few cases we have mounted *Bad News* in ways that deviate from some of the above description. In this section, I will describe these variants.

Augmented Radio Play

In a few cases, we have mounted *Bad News* itself into a larger media experience that features a live audience. We call this configuration an *augmented radio play*, and it tends to work as follows. First off, the gameplay area is secluded in a private room somewhere, but the wizard command center is housed on a stage in an auditorium area. In this area, all of the various interfaces are displayed on a large screen on stage: wizard console, actor–wizard chat, actor interface, and player interface. Figure 10.6 shows an example of how this might look.

Before we start the performance, codesigner and guide Adam Summerville selects a willing player from the audience, and the guide preliminaries then occur on stage (so that the audience also learns how the experience works). Next, the player is taken to a secluded area that is removed from both the auditorium and the gameplay area. At this point, the wizard generates a new town and the wizard and the guide explain the purpose of story sifting, which is then carried out on stage through livecoding.[36] As part of this process, we solicit audience members to call out ideas about interesting story material that the wizard may attempt to excavate; in many cases, these audience call-outs have significantly altered the

[36] In this case, the wizard's livecoding is a kind of performance of its own, a phenomenon about which Nick Collins has written [200].

Figure 10.6: *Bad News* being performed as an *augmented radio play* at a packed theatre in San Francisco. In this experience variant, we seclude the gameplay area in a private room and set up the wizard command center on a stage in an auditorium area. Here, before a live audience, all of the various interfaces are displayed on a large screen on stage: wizard console, actor–wizard chat, actor interface, and player interface. With a willing audience member as player, the performance then proceeds in the private room, but critically audio from there is piped into the auditorium, so the wizard and guide and audience members can hear what is happening. As the performance transpires, the wizard and guide provide commentary and solicit audience suggestions about ideas to communicate to the actor. In this way, the audience experience resembles that of hearing a radio play, though in this case it is augmented by a look at how the experience works behind the scenes. This shot was taken at a 2017 performance at PianoFight, a theatre in San Francisco; just out of frame is Adam Summerville, who held the other microphone and emceed the event.

shape of the performance. Finally, the player is led into the gameplay area and the mortician scene begins. Critically, however, the audio being captured by the microphone there is piped into the auditorium, so the wizard and guide and audi-

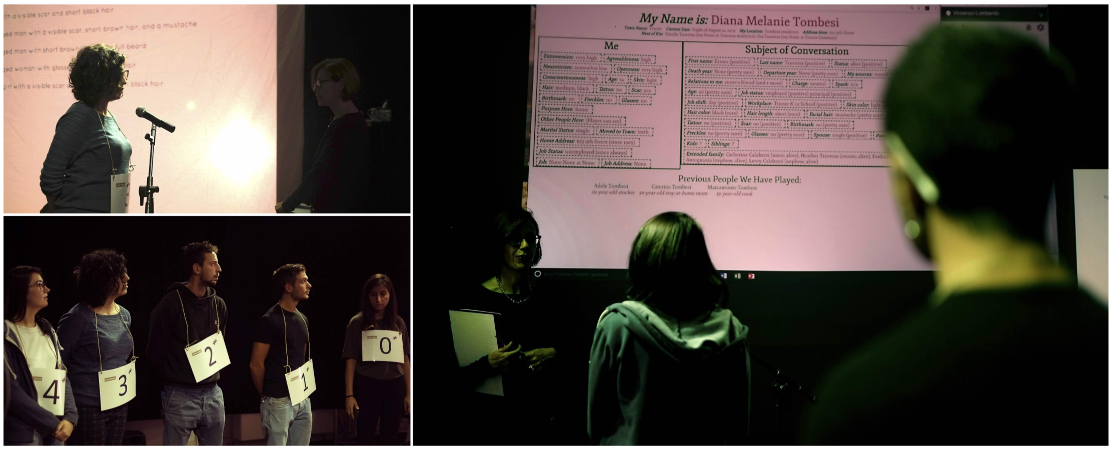


Figure 10.7: Scenes from *Cattive Notizie* (2017), a *Bad News* variant housed in a *CAVE*-like physical space. I developed this one-off piece with researchers Vincenzo Lombardo and Antonio Pizzo during a visit to the University of Turin in Italy. In the variant, the gameplay area (which is typically anchored by a model theatre) was reconstituted as an entire room whose four walls were covered in mounted screens—as such, the result worked similarly to a *CAVE* experience. Instead of a single actor, we enlisted several dozen, each of whom could be called upon to act over the course of a playthrough. To speak to a particular character, the player indicated the number of the corresponding actor (lower left), and the two positioned themselves on markers specifying where exactly to stand for the conversation (top left). From the vantages afforded by these positions, an actor could see the actor interface displayed on one wall (right), while the player could not (though she could still see her player interface). While the project was very experimental and the actors did not have enough time to develop the peculiar set of skills required for computationally assisted performance, the results were highly intriguing.

ence members can hear what is happening. In this way, the audience experience resembles that of hearing a radio play, though in this case it is augmented by a look at how the experience works behind the scenes.

Cattive Notizie

Additionally, there has been another more radical reformulation of *Bad News*, but one that takes a different name. In the course of a visit to the University of

Turin in Italy, in October 2017, I collaborated with researchers Vincenzo Lombardo, Antonio Pizzi, and Rossana Damiano to design and mount a *Bad News* variant that we called *Cattive Notizie* (Italian for 'bad news'). This was the group who had produced *Hot Bread* [238], the *Talk of the Town*-driven work of curationist emergent narrative that I discussed above.

In this experience, the gameplay area was reconstituted as an entire room whose four walls were covered in mounted screens—as such, the result worked similarly to a *CAVE* [229, 228] experience. Instead of a single actor, we enlisted several dozen undergraduates who would each potentially act over the course of a playthrough. To enter a new location in the town, the player carried out the physical action of entering the *CAVE*-like room, where an array of actors stood on numbered circles corresponding to the numbering of nearby characters shown on the player interface, as Figure 10.7 shows. On one of the walls facing the player, we displayed the player interface. To speak to a particular character, the player indicated that she would like to converse with the person of the corresponding number, and the two positioned themselves on markers specifying where exactly to stand for the conversation. From the vantages afforded by these positions, the actor could see the actor interface displayed on one wall, while the player could not (though she could still see her player interface).

Aside from these alterations, the experience worked in the same way, except that I modified aspects of the *Talk of the Town* simulation instance so that each storyworld would be rigged to be an Italian–American town. While the project was very experimental and the actors did not have enough time to develop the peculiar set of skills that such a performance requires, the results were highly intriguing. Generally, I am very excited about the prospect of a *Bad News*-like piece that distributes the experience across a more extensive physical space with multiple actors and multiple players, perhaps even employing multiple wizards.

A New Troupe and the Home Version

Finally, I would like to note the ongoing efforts of Jonathan Lessard and performer Noah Drew to mount *Bad News* on their own, with Lessard as wizard and Drew as actor. This project is still in a preliminary phase, as its team takes on the challenges of learning how to both interact adeptly with someone else's simulation engine and also develop a rapport in the peculiar area of computationally assisted performance. All of the *Bad News* codesigners are very excited about the idea of *Bad News* being performed by people who are not us, and indeed this project relates to our long-held ideas about a so-called *home version* of the experience that would enable anyone to put it on informally with their friends as a kind of computationally driven *tabletop storygame* [984, 1333]. In his dissertation, Ben Samuel briefly describes our ideas for a home version [1084, p. 383], which Adam Summervile is now planning to pursue as part of his early research agenda as a professor at Cal Poly Pomona, a position he has just started this fall.

10.3 A Curation Example

The following example of *Bad News* curation transpired in the course of a performance at the Fort Mason Chapel in San Francisco on September 30th, 2018, where *Bad News* was installed as part of *Come Out & Play San Francisco*, a festival of public games. In this installation, the gameplay area was a private room closed off from the rest of the chapel, and so I carried out my wizard duties from an entirely separate area; like in all our performances, I could still listen in via a microphone in the gameplay area. For this particular playthrough, we recorded audio and also saved copies of the actor–wizard web chat and wizard terminal session log; the following quotations originate from those sources. None of them are edited in any way, though of course they are excerpts taken from the larger experience duration. In these quotations, the speaker of each utterance is indicated by a prefix and excerpts from my wizard livecoding are formatted as Python code listings; in the listings, commands are preceded by '>>>' and the lines following a command show the returned values.

Initial Story Sifting

As I explained in Section 10.2.4, a *Bad News* performance begins with the wizard (me) carrying out a process of initial *story sifting*, whereby a set of narratively potent emergent scenarios may be identified for delivery to the actor. Before I can do this, however, I need to find out who the deceased character and next of kin are, and then relay that information to Ben. At the start of a new performance, neither of us know anything about the storyworld, because it did not even exist until that moment. In this performance, a young deli cashier named Ronald Zeise has died, and there are a record *twelve* next of kin:[37]

```
>>> print d
Ronald Zeise, 1954-None
>>> print d.occupations
[<occupation.Cook object at 0x10cf327d0>,
<occupation.Cashier object at 0x10da72450>]
>>> print d.occupations[0]
Cook at Zilencio 1973-1975
>>> print d.occupations[1]
Cashier at Steele Creek Avenue Delicatessen since 1978
>>>
>>>
```

[37] As I noted above, in *Bad News* the next of kin is the closest living familial relation in town, and if there are several characters related to the deceased character by the same degree, then each is treated as a next of kin. Notifying any single one of these individuals suffices. In the code examples in this section, the variable `d` is bound to the deceased character, while `nok` is bound to a list containing all of the next-of-kin characters.

```
>>> print nok
[<person.Person object at 0x109da2fd0>,
<person.Person object at 0x1093c3450>,
<person.Person object at 0x1095da890>,
<person.Person object at 0x1070cf910>,
<person.Person object at 0x108b93990>,
<person.Person object at 0x10b4d5e90>,
<person.Person object at 0x108e3e450>,
<person.Person object at 0x10da86250>,
<person.Person object at 0x1098b7650>,
<person.Person object at 0x10af46350>,
<person.Person object at 0x106dd5c90>,
<person.Person object at 0x10762c7d0>]
>>> len(nok)
12
>>> for n in nok:
...     print d.relation_to_me(n)
...
cousin
cousin
cousin
aunt
cousin
cousin
cousin
cousin
cousin
cousin
aunt
uncle
```

This was a remarkable number, and in fact it was the first case of a deceased character with extended family as the next of kin—more typically, this group comprises fewer and more immediate family members. Having sifted out this pertinent information, I hurriedly relay it to Ben via our actor–wizard web chat:

WIZARD: d is Ronald Zeise, 1954-

WIZARD: Cook at Zilencio 1973-1975

WIZARD: >>> print d.occupations[1]

WIZARD: Cashier at Steele Creek Avenue Delicatessen since 1978

WIZARD: WHOA

WIZARD: 12 NOK

ACTOR: WHAT

At this point, the player is still in a preliminary session with the guide, who is explaining how the experience works. Upon further hasty exploration, I encounter an interesting nugget—the deceased's great-great-grandfather was a founder of the town who had worked at night as the cemetery groundskeeper for nearly fifty years:

```
>>> print d.father
Edward Zeise, left city in 1973
>>> print d.father.father
Robert Zeise, 1885-1954
>>> print d.father.father.father
None
>>> print d.mother
Carol Zeise, left city in 1973
>>> print d.mother.father
None
>>> print d.father.mother
Hattie Horsman, 1892-1961
>>> print d.father.mother.father
Henry Horsman, 1840-1909
>>> print d.father.mother.father.father
Walter Horsman, 1817-1886
>>> print d.father.mother.father.father.occupations
[<occupation.Groundskeeper object at 0x1038fcd90>]
>>> print d.father.mother.father.father.occupations[0]
Groundskeeper at Mosses Cemetery 1839-1883
>>> print d.father.mother.father.father.occupations[0].shift
night
```

As I am looking this up, Ben inquires about the deceased character's family and in my response I communicate that this emergent scenario could make for an interesting story for Ben to deliver conversationally at some point:

Actor: what happened to parents

Actor: clearly no spouse

Actor: no siblings I gues?

Wizard: they left town in 1973

Wizard: siblings left too

Wizard: they were younger

Wizard: they are all descendents of Walter Horsman, 1817-1886

Wizard: founder

Actor: nice

Wizard: groundskeeper of the cemetery 1839-1883

Wizard: i like a story about him being a creepy guy?

Actor: coooool

Actor: yeah

Wizard: it was night shift

Actor: perfect

Wizard: old man Horsman

Wizard: lol

Actor: founded the town

Actor: just so he could be a gravekeeper

Wizard: hahaha yes

WIZARD: love it

Shortly after this exchange, the player is led into the gameplay area and the mortician's scene commences. This means Ben is not available to continue chatting, but this scene does not require maintenance of the player and actor interfaces, which allows me to continue story sifting without hindrance. In the course of searching more through the emergent material pertaining to the deceased's ancestors, I encounter a remarkable case of a restaurant named Zilencio that had operated for 131 years but recently had shut down (recall that the year is 1979):

```
>>> print list(d.ancestors)[2].occupations
[<occupation.Waiter object at 0x103d52cd0>,
<occupation.Farmer object at 0x104d24b50>]
>>> print list(d.ancestors)[2].occupations[0]
Waiter at Zilencio 1850-1875
>>> print list(d.ancestors)[2].occupations[1]
Farmer at Thomas Dowey's farm 1875-1894
>>> print list(d.ancestors)[2].occupations[0].company
Zilencio, 317 Steele Creek Avenue (1844-1975)
>>> zil = list(d.ancestors)[2].occupations[0].company
>>> print zil
Zilencio, 317 Steele Creek Avenue (1844-1975)
```

I now become curious about the circumstances of Zilencio's recent closure, so I sift further to see who owned it last:

```
>>> print zil
Zilencio, 317 Steele Creek Avenue (1844-1975)
>>> print zil.owner
Proprietor at Zilencio 1973-1975
>>> print zil.owner.person
Douglas Salva, 31 years old
```

Apparently a character named Douglas Salva was Zilencio's final owner, and he had only been in charge for two years before its remarkable closure. Digging deeper, I learn that Salva had originally been a waiter at the restaurant, but was now a stocker at a department store:

```
>>> doug = zil.owner.person
>>> print doug
Douglas Salva, 31 years old
>>> print doug.occupations
[<occupation.Waiter object at 0x10bdef5d0>,
<occupation.Proprietor object at 0x10ce79410>,
<occupation.Stocker object at 0x10d915910>]
>>> print doug.occupations[0]
Waiter at Zilencio 1966-1973
>>> print doug.occupations[1]
Proprietor at Zilencio 1973-1975
>>> print doug.occupations[2]
Stocker at Flom Department Store since 1977
```

I wondered why a waiter would become the owner of a restaurant, since I know that in-house promotion in *Talk of the Town* tends to elevate employees with a occupational status immediately below the open position (in this case, a restaurant manager). As I explained in Section 9.2.7, however, in addition to favoring current employees for open positions, characters also prefer to hire family members, and so candidates meeting both criteria score highly in the utility procedure that drives the hiring process. As such, I now suspect nepotism, and my sifting confirms this:

```
>>> print doug.occupations[1].preceded_by
Proprietor at Zilencio 1964-1973
>>> print doug.occupations[1].preceded_by.person
Henry Ereth, 69 years old
>>> henry = doug.occupations[1].preceded_by.person
>>> print henry.relation_to_me(doug)
nephew
```

More intriguingly still, I find a connection to the deceased character. Apparently the late Ronald Zeise was a cashier at a deli that now stands on the former site of Zilencio, and moreover he had actually been a cook at the latter during the brief period of Salva's ownership:

```
>>> print zil.lot
A lot at 317 Steele Creek Avenue on which Steele Creek
Avenue Delicatessen has been erected
>>> print d.occupation
Cashier at Steele Creek Avenue Delicatessen since 1978
>>> print d
Ronald Zeise, 1954-None
>>>
>>>
>>> print d.occupation
Cashier at Steele Creek Avenue Delicatessen since 1978
>>> print d.occupations
[<occupation.Cook object at 0x10cf327d0>,
<occupation.Cashier object at 0x10da72450>]
>>> print d.occupations[0]
Cook at Zilencio 1973-1975
```

This seems like a potent emergent scenario, especially given the connection to the deceased character, and so I relay it to Ben with the suggested narrative spin that the town is angry at Salva for killing a beloved institution:

WIZARD: there was a town instutition

WIZARD: Zilencio, 317 Steele Creek Avenue (1844-1975)

WIZARD: that closed after 130 years

WIZARD: b/c of this failure Douglas Salva, 31 years old

WIZARD: who now works at dept store

WIZARD: as stocker

ACTOR: Who was Salva before?

WIZARD: whole town is mad

WIZARD: owned Zil

Curation at Work

By now, the mortician scene has ended and the player leaves the death scene and moves into the lobby of the deceased character's apartment complex. Here, he initiates a conversation with a janitor there named Clarence and explains that he is a restaurateur who is working to open a dinner theatre in the town. After a few minutes, the player asks if the janitor knows any town lore:

> PLAYER: One of the things that really puts a town on the map for us is a sense of local lore. Local legend. We had a really successful restaurant for a while, in another state. One of the reasons it was so great is because the town had its stories.
>
> ACTOR: Yeah, uh-huh.
>
> PLAYER: You know, it wasn't all pretty. We did have the, you know, we had the feud between, you know, the different families. And I was wondering if you felt like this was a candidate town for that and, uh, if you could tell me anything.

This is a perfect opportunity for Ben to impart the two emergent scenarios that I have uncovered—the creepy cemetery groundskeeper Walter Horsman and the late town institution Zilencio—and so I quickly remind him in the chat by hastily copying and pasting information from my Python terminal:

> WIZARD: Zilencio, 317 Steele Creek Avenue (1844-1975)
>
> WIZARD: Douglas Salva, 31 years old
>
> WIZARD: >>> print zil.owner
>
> WIZARD: Proprietor at Zilencio 1973-1975
>
> WIZARD: Walter Horsman, 1817-1886
>
> WIZARD: Groundskeeper at Mosses Cemetery 1839-1883

Ben then launches into a brilliant turn of conversational storytelling that (as I will explain shortly) nicely demonstrates our narrativization motto, *augment but do not contradict*:

> Actor: Yeah, we got a couple stories. I got two stories for you.
>
> Player: Okay.
>
> Actor: I've got an old story and I've got a new story.
>
> Player: Great.
>
> Actor: Old story first.
>
> Player: Okay, let's do old.
>
> Actor: Okay. The town was founded, I don't know, hundred years ago—hundred fifty years ago—something like that. Guy who founded it, rumor has it, the reason why he wanted to found the town in the first place—because he just wanted a place where he could be a grave keeper.
>
> Player: Huh.
>
> Actor: He founded the town.
>
> Player: A grave keeper.
>
> Actor: He could have any job he wanted. Next thing, very next thing, starts up a cemetery, becomes the undertaker there.
>
> Player: Can't imagine he'd have a whole lot of business to begin with.
>
> Actor: [laughs]
>
> Player: I guess if you build it they will come.
>
> Actor: I suppose so. Or maybe he had some more sinister plans in mind. Maybe he wanted to accelerate the cemetery a little faster. Maybe he built the town to attract flies into his trap.

Player: That certainly is a chilling thought, Clarence.

Actor: [laughs] Ah, I mean, it sounds crazy, right? No way. No way that can be true. [laughs] But we still, we still talk about him! Walter Horsman is his name.

Player: Walter Horsman?

Actor: Old Man Horsman. I mean he's long dead. [laughs] His body in the very cemetery that he started.

Player: Has the cemetery become quite liked by, uh—

Actor: Yeah, yeah, it's doing okay. Yeah. [sighs] Old Man Horsman.

Player: I have to say that I sort of understand the desire to devote one's life to the next world.

Actor: What? What do you mean?

Player: I just find it interesting that, you know, someone can be alive at one minute and the next thing you know, they're gone. Like you just told me.

Actor: I feel like you just summed it up.

Player: Yeah.

Actor: I feel like that's about all there is.

Player: That's about—that's life! I agree.

Through the delivery of this story about the creepy gravedigger, Old Man Horsman, a procedure of curation has been consummated. Critically, this procedure has worked according to the curationist architecture introduced in Section 5.3. First, the history of a *simulated storyworld* transpired. This was a *Talk of the Town* town named Mosses, founded in 1839, and one of its settlers was a man named Walter Horsman who came there to work in its newly established cemetery as a night-shift groundskeeper. Horsman then proceeded to live out a life in the town: he worked the same job for the next forty-four years, married and had children, and then passed away in 1886. (Incidentally, the restaurant Zilencio was opened in 1844, which means he likely visited it many times over the course of his life.) Over the next century, his children had children who had children and so forth, and one of his great-great-grandchildren was a man named Ronald Zeise who died suddenly in the summer of 1979.

All of this information was recorded by *Talk of the Town* in the form of stored data that could be queried in an interactive Python session—in curationist terms, this stored data is a *chronicle*. In my capacity as wizard, I then began working as a *story sifter* to excavate material capturing all the information that I have just outlined. I encountered this particular material by working backward through the deceased character Ronald Zeise's family tree, and I decided to sift it out because the notion of a night-shift gravedigger is evocative, especially in the context of *Bad News*. Next, I packaged up this sifted material and passed it on to the actor, Ben Samuel, who worked as a *narrativizer* to turn the raw material into an actual narrative artifact that is delivered through conversational storytelling. Through this delivery, the narrative artifact was *mounted* in a full-fledged *media experience* that was then encountered by an *experiencer*. In this case, this experience was *Bad News* and its experiencer was the player, who engaged with the mounted narrative artifact through conversational interaction in the context of the experience.

Augment But Do Not Contradict

In Section 10.6, I will again outline the curationist configuration that underpins *Bad News*, but for now I would like to include a few more examples of curated emergent stories that were also mounted in the course of this particular perfor-

Figure 10.8: *Bad News* actor Ben Samuel converses with a pair of players during a performance at the San Francisco Museum of Modern Art. Behind the fixed lower curtain of the model theatre is a hidden screen displaying an actor interface and a chat session with the wizard. Ben curates the raw material conveyed through these channels to build narrative artifacts recounting *Talk of the Town* emergent scenarios, which he delivers to the player(s) through conversational storytelling.

mance. The following exchange begins right where the last one left off, as part of the same conversation with the janitor in the lobby:

> Player: So what about this new story?
>
> Actor: Ah, okay, okay. Well, the tension is still a little high around town about this. So you might want to be a little careful about who you talk to. Ah, have you heard of Zilencio?
>
> Player: Zilencio? No, I haven't. Sounds promising.
>
> Actor: Uh, yeah. It was an institution.
>
> Player: Oh.

Actor: It's been a part of our town since as long as I can remember. I remember my parents and my grandparents telling me about it. It's gone.

Player: Really? What kind of institution was that, Clarence? Is it an insane—

Actor: Oh, no, no, no. By 'institution', I mean a—by 'institution' I mean a—what's the word? Landmark.

Player: Oh, landmark, oh. A place that people knew.

Actor: Yes. Yeah, yeah, yeah, sorry about that.

Player: A destination, if you will. What was it? Was it a business?

Actor: Oh, restaurant. It was a restaurant. So you—you, doing your research—are gonna wanna hear about this. And specifically, you're not gonna wanna hire Salva.

Player: Salva? Is Salva one one of the staff there?

Actor: Douglas Salva.

Player: Okay.

Actor: Waiter there for ten years. Became a proprietor of it—*the* proprietor of it—and within two years this destination, if you will, closed down.

Player: Really?

Actor: Gone.

Player: Why, what happened? Just poor management? I know that feeling.

Actor: Yeah! Yeah, you know what? You know, Douglas might want to talk to you. You know, he might be able to learn a thing or two from you. I don't, I don't know anything. And frankly, I don't want to talk to the guy—everyone's pretty mad at him.

Again, the same procedure as above has been carried out: the raw transpiring of a storyworld has been captured in a chronicle, which I sifted to extract material that was delivered to Ben, who narrativized that material to build a story, which he then mounted into the experience through conversational storytelling. Note that while Ben's story *augments* the raw source material—data expressing that there was a restaurant that operated for 131 years until it was closed shortly after being taken over by a character named Douglas Salva—it does not *contradict* it. This distinction is critical with regard to the artistic orientation of the curationist framework that is the subject of this dissertation.

As I have argued at numerous points in this dissertation, following the ideas of Hayden White and Arthur Danto and Espen Aarseth and others, stories themselves cannot manifest in records or in the raw transpiring of a simulation. Rather, emergent stories come to actually exist through a procedure of curation. Because this procedure is constructive, it is therefore also augmenting. In fact, augmentation is unavoidable in the narrativization step, if only for the simple reason that narrativization requires the construction of what Arthur Danto has called *narrative sentences* [239]. As I explained in Section 3.1.1, a narrative sentence is one that packs descriptions of multiple events (and their interrelations) into a single sentence, such as the canonical "The Thirty Years War began in 1618" [239, p. 155]. Even the construction of a narrative sentence is an augmenting act, since crafting one requires the assembly and aggregation of multiple chronicle entries into a single unit of content. This means that narrativization is inherently steeped in a process of augmentation, and as such, augmentation is not treated as an offense in the curationist framework.

In emergent narrative, a simulated storyworld will always be impoverished in its representation (due to the ontology of simulation), and so if a curated story is to feature the level of detail that is typical of narration, augmentation likely must occur. The key, however, is to not *contradict*: when the facts of a chronicle are contradicted, the history of the simulated storyworld as it actually transpired is polluted, and narrative intervention has occurred thereby. As I argued in Section 5.1, intervention kills the aesthetics of emergent narrative, and so contradiction should be avoided in the narrativization process.[38] Thus, while Ben's narrativization augments the sifted source material to inject flavor and detail, his process is still fully in line with the curationist approach. In fact, his method exemplifies the very kind of narrativization that I now seek to do automatically, starting with *Sheldon County*.

The Story Retold

Following his conversation with the janitor, the player inquires with the landlord of the apartment complex and asks for directions to the Steele Creek Avenue Delicatessen. As I noted above, the late deceased character worked at this deli,

[38]To be clear, I think it could be fruitful to deliberately model a *subjective narrativization* that *does* contradict the storyworld, but in ways that are specifically targeted to a subjective storyteller. While an example of this does not appear in the transcripts printed in this section, a *Bad News* character may embellish in a story, or even tell outright lies, if doing so makes sense given her relation to the people and events being recounted. This entails narrativization that contradicts the simulation, but in ways that deliberately target a subjective telling. Moreover, such subjectivity will be readable to the player in that she is likely to assume that not all the stories she hears are true, as in the real world. To name another example, my original dissertation topic concerned characters who would generate self-serving conversational stories recounting past events. But in a project like *Sheldon County*, a generative podcast about the emergent events of a uniquely simulated storyworld, I would not target a subjective telling because its stories are told from the vantage of an omniscient narrator. If its underlying system, *Sheldon*, were to contradict the storyworld through its narrativization, the project itself would be undermined—what is the point of simulating a storyworld if the telling will rely on artifice and invention, rather than the actual material generated by that world? Indeed, when contradiction is permitted, there is no point to taking the approach of emergent narrative, and in fact, the result will not even be a work of emergent narrative. To be clear, this does not mean that such a method is inferior, but rather that it is something entirely different. As I have already explained, this thesis is intended primarily as an art manifesto that promulgates a fiercely emergentist approach to procedural narrative. My value-laden language is meant to instantiate this artistic orientation, not to cast judgement on intellectual persuasions or technical methods that differ from mine.

which was incidentally on the site of the former town institution Zilencio. Here is an excerpt from this conversation (note the parenthetical wizard chat messages that I have injected at approximately the points at which Ben would have received them on his hidden actor interface):

> Actor: I think you might be able to find your way there. It's, uh, well it used to be one of the most popular places in town. [laughs]
>
> Player: I heard it closed recently. Are you talking about Zilencio?
>
> Actor: Yeah. Steel Creek has yet to prove itself. Give it another 130 years!
>
> Player: Wow, well I haven't heard all the stories, but I sure would love to hear that story.
>
> Actor: Have you heard about Doug Salva?
>
> Player: Douglas Salva? I heard he was the proprietor.
>
> Actor: Yeah, yeah. If you could call it that.
>
> Player: Not that I'd want to probably hire him for a restaurant that I'd create, would I? But I've heard there's some, uh, controversy about him. About how he ran the place.
>
> Actor: Don't think there's a lot of controversy. Uh, the place existed and now it doesn't. Pretty, uh, open-and-shut case.
>
> Player: Ah, I see. Well, do you think I could talk to Douglas? I sure would love to include his story in, you know, a piece I'm putting together.
>
> [Wizard: Stocker now]
>
> [Wizard: at Flom Dept Store]
>
> Actor: [laughs] Yeah, yeah. Uh, you can probably find him—
>
> [Wizard: LOL]

> [Wizard: he lives there!]
>
> [Wizard: unit #7]
>
> Actor: Well, there's two places you might be able to find him. The Flom Department Store had the decency to take him in after, you know, he suddenly found himself in need of a job. Uh, but the sad sack actually lives right here. Unit #7. You might be able to find him.
>
> Player: Oh, really?
>
> Actor: Yeah.
>
> [Wizard: he's at home]
>
> [Wizard: works nights]
>
> Player: Wow, Douglas Salva lives right here?
>
> Actor: Yeah. Don't remind me. I have people picketing the place more often than I care to count.

Here, we find a case of Ben curating the same sifted material into an entirely new narrative artifact, which is likewise mounted into the experience through conversational storytelling. This is achieved by retargeting the narrativization procedure so that the result is a conversational story delivered from the perspective of, and in the style of, a different character than before (the landlord, as opposed to the janitor). While he does not contradict the storyworld—in fact, he references the additional detail that Salva now works at the Flom Department Store—there has been a retargeting to the variant perspective of the landlord character. Relative to the janitor, this character's personality profile led to a more brusque performance, and the telling differs accordingly. Also in this example, we see what might be called *just-in-time story sifting*: a primary task of the wizard is to provide the actor information that is not expressed on the actor interface but is needed in the conversation. Ben is very good at parsing these messages quickly and naturalistically—many players have been surprised to learn that he relied on a hidden screen—but I still try to keep these messages to a minimum.

The Man Himself

As his next move, the player buzzes Douglas Salva's apartment. Before this critical scene starts, I send Ben some additional sifted material to guide his performance (and his narrativization):

> WIZARD: okay you're doug
>
> WIZARD: very open
>
> WIZARD: very neurotic!
>
> WIZARD: you were a great waiter from 66-73
>
> ACTOR: gonna invite him in ok
>
> WIZARD: okay
>
> WIZARD: Ronnie is a neighbor
>
> WIZARD: very high charge
>
> WIZARD: uncle gave you job
>
> WIZARD: after you were greater waiter for decade
>
> WIZARD: Henry Ereth, 69 years old
>
> WIZARD: your great uncle owned it before that
>
> WIZARD: for 30 years
>
> WIZARD: henry ereth
>
> WIZARD: before that great great uncle owned

> WIZARD: it's a family business
>
> WIZARD: uncle Henry never had kids, left it to you

Salva answers the buzz, but does not say anything, and so the player speaks up to ask if anyone is there, and a conversation transpires:

> PLAYER: Uh, Mr. Douglas Salva? This is Winston—
>
> ACTOR: Just—I don't, I don't wanna talk to anyone, okay? Leave me alone.
>
> PLAYER: Oh Mr. Salva? I, um, I must have misrepresented myself. I'm a restaurateur, um, looking to start up a restaurant. I actually wanted to make a proposal for you.
>
> ACTOR: You don't, you don't want me near your restaurant. Is this some kind of joke?
>
> PLAYER: No, I cannot be serious enough when I tell you that this is no joke whatsoever, Mr. Salva. Um, do you think I could have the opportunity to, uh, to introduce you to an idea. If you don't like it, you can send me on my way. This is not a joke and I am not here to humiliate you.
>
> ACTOR: [sighs] Okay. Yeah. Yeah, why not. Uh, uh, come in. Come on in.

Salva buzzes the player into the apartment (signaled by Ben drawing open the curtain of the model theatre) and Ben proceeds to curate the emergent Zilencio material once more, but this time from the dejected perspective of Salva himself (and using the additional material that I sent to him over the chat):

> PLAYER: Is this Mr. Salva?
>
> ACTOR: Call me, call me, call me Doug.
>
> PLAYER: Winston Delacroix.

Actor: Hi.

Player: Hi. Hi. I'm from Newtown. And I would like to make a proposal to you. I do appreciate you letting me in. I hear there's, um, quite a lot has gone under the bridge in town recently. Um, but, as my pappy used to say, there's nothing like learning from one's failures to get up and, um, and do a great job. Um, one of my business school professors once told me, if I haven't failed before—if I haven't messed up—then he's not going to listen to me, because it's only through mistakes and failure that we actually learn how to proceed.

Actor: Yeah, look. Look. Thank you. Thank you. That means a lot to me. It does. It really does. Um, I feel your, your teacher who told you that in a, in a nice classroom—like, that sounds really inspirational. That sounds, that sounds really great. Um, imagine, if you will, that the failure is at a restaurant that you've been a waiter at for ten years. The best waiter they've ever seen. A waiter so good they made you the boss. They made you the boss. [sobbing] Everyone loves you. Everyone loves the restaurant. [bawling] And I did my best. I'm sorry. I'm sorry. I'm, I'm—you're the first person that hasn't yelled at me in so long. I'm so sorry. I'm so sorry.

Player: Mr. Salva, you need not apologize for one instant. Someone with your mettle and your experience is exactly the kind of person we're looking for. Someone who can really feel and understand what can happen.

Actor: [sobbing] Yeah.

Player: Now, this might sound bizarre to you. This might sound strange. But we are looking for someone exactly like you to help us with the new restaurant we want to start. And I know—I know that's gonna sound strange—but one thing we need to start with is honesty. So I need you to tell me in about a sentence or two what went wrong. And then when we can meet on that level, then we can take it to the next step. Because I believe it's only by admitting—it's only by putting it out there what happened—that we can have a true bond. Would you do that with me, Mr. Salva?

Actor: [sobbing] Yeah.

Player: Okay, excellent. Tell me what happened.

Actor: It's easy. I, uh, I rose above my station. Simple as that. I thought that like, you know, I just knew it all inside and out. I thought, I thought that I could make a difference. I thought that I could take this great place that I loved so much and just make it even better. But I was naive. I was overly optimistic. I lacked any formal business training. I clearly had no intuition for it. I convinced the powers that be—I convinced my uncle—to give me the job. And I wasn't ready. I didn't know what I was doing.

Player: Mr. Salva, so I do follow along: who is your uncle?

Actor: Huh?

Player: Who is your uncle?

Actor: Um, um, Henry.

Player: Henry, okay. Is that—okay, I'm sorry, keep going. I didn't mean to slow you down.

Actor: I thought that I could—you know, he meant so much to me, the restaurant meant so much to me—I thought that I could follow in his footsteps. Keep it in the family. I just didn't know what I was doing. Um, I wish, I wish that wasn't the truth. I wish that I could tell you that, you know, people didn't believe in me, and because of that they weren't working hard for me. I wish I could tell you that there was some problem with the economy and that all of the restaurants were suffering. No, it's nothing. Everyone believed in me. The economy was fine. It was me.

Player: So what happened?

Actor: I just, I spent too much. I didn't—I spent money on—I spent money on I don't know what you would call it. See, I don't know what you would call it. Marketing, I guess. Advertisements. I tried, I tried to change the feel of the restaurant. I wanted—people already had such high respect for it—I wanted to make it classier. I wanted to make it highfalutin.

Player: Understood. I understand now.

Actor: I had, I had ice sculptures. I had these giant statues. I imported pillars.

Player: Zilencio was the name of the restaurant, as I understand it.

Actor: Yeah.

Player: You did not hold back, did you? And what did your uncle think of this, when you were spending all this money?

Actor: He never, uh, he never told me. He just said, uh, you know, I trust you.

Player: Do you have any other family that were impacted by this?

Actor: I mean, I think, in general, the town has been understanding enough that it's my fault—and my fault alone—that it happened. I don't think my family has been under fire, as it were. Thank goodness. It could be worse.

From here, the player proceeds to concoct a proposal for a new restaurant in town that would actually be themed around the redemption and resurrection of Douglas Salva: "I should tell you the restaurant name we're thinking of, or playing with, is 'The Rise of the Phoenix"'.[39] Through discussion of whom to invite to a press event, the player cleverly discovers the identity of the next of kin.

The Importance of Curation

In this way, the story material that I sifted out of the chronicle prior to the start of the performance was curated into a story that not only was mounted in the experience, but moreover came to form its bedrock. The Zilencio story became the structural scaffold of the entire performance, one whose centrality actually

[39]Of course, this was not a real plan, since the player character was in actuality a mortician's assistant, not a restaurateur. In the epilogue following gameplay, we discovered what became of Douglas Salva in the following decades, as I will explain in the next section.

made the deceased character into something more like a side character. Recall, however, that this material was only encountered through an improvisational exploration of the storyworld's history. As the terminal excerpt above showed, I was serendipitously exploring the work history of one of the deceased's characters ancestors, when I encountered Zilencio and its remarkable duration of operation. This was only one of the storyworld's emergent scenarios, though, and I could just as well have sifted out material pertaining to another, which in turn would have utterly changed the course of the performance.[40]

As such, this is a clear expression of the importance of curation in *Bad News*, as well as the tendency of the underlying *Talk of the Town* simulation engine to produce narrative cornucopias, as I explained in the last chapter. One consideration that we have often discussed, and that players have likewise brought up, is all of the myriad emergent scenarios that we never even come to know about in the course of a performance. Who knows what other intrigue lied hidden in the town of Mosses? Indeed, this is part of the appeal of automatic curation: a computerized story sifter could sift through *all* of the material stored in a storyworld chronicle to identify not only the most promising nuggets, but also the inscrutable connections that serendipitously bind those nuggets. In Chapter 12, I return to the notion of automatic curation, which I am now exploring in *Sheldon County*.

10.4 The Pleasure of *Bad News*

What is the pleasure of *Bad News*? In this section I will attempt to answer this question, partly through the utilization of the intellectual framework that I

[40] As I noted above, in our so-called *augmented radio play* configuration, we encourage audience members to call out for ideas as to what to look for in story sifting. In this way, the audience becomes a collaborator in the curation procedure, and in some cases audience suggestions have significantly altered the experience.

developed in Chapter 3 (to appraise the very form of emergent narrative). Additionally, I will argue for the pleasure of this work in a more essential way, by appealing to its reportable successes, including awards and favorable media attention. In the case of my appraisal of *Diol/Diel/Dial*, this was not possible—since it appears that nobody but me has experienced that piece, which should itself be viewed as a damning appraisal—but in my view this is the strongest form of evaluation for projects in the area of computational media. I think it is worthwhile to make purely technical contributions, but my personal aim is to develop compelling media experiences, and the only way to evaluate such experiences is to release them into the world to see what others think.

Evidence of Success

Let us start with the success of *Bad News*, which suggests that it does indeed give pleasure (that I will in turn attempt to unpack in the remainder of the section). First, *Bad News* has been installed internationally, at a number of venues, and in that course it has won awards. One of our first installations was at the 2016 ACM CHI conference, where we took home the Innovative Game Design award in the student game competition. Later that year, the project was accepted into IndieCade, the foremost international festival for independent videogames, where *Bad News* won the Audience Choice award. Additionally, we have installed the piece at the Slamdance Film Festival, where it was included in the DIG exhibition of digital works, and the San Francisco Museum of Modern Art, as part of an exhibition of *mixed reality* [894] works. *Bad News* has also been performed outside the United States, following invitations from universities in Montreal, Canada, and Turin, Italy.[41]

[41] Specifically, it was installed at Concordia University, Université de Montréal, and the University of Turin, though in the latter case this was specifically the experience variant *Cattive*

The project has earned favorable critical reception, as well, both in the media and among renowned game designers. Writing for *Rolling Stone*, Steven T. Wright proclaimed, "this marvel of procedural performance can only be played by a lucky few, and that's a crying shame" [1359, n.p.]. Game designer Mattie Brice listed it among her top ten games of 2016 [137], and designer and scholar Tracy Fullerton called it "a game that shook my world".[42]

While performing *Bad News* is gratifying in and of itself—and while we believe it makes technical and intellectual contributions—its various successes in terms of invited and juried installations, favorable reviews, and festival awards are the accomplishments that have made us most proud. In turn, I contend that such accolades provide the strongest form of positive evaluation that can be reported in the field of computational media.

Actual Worlds and Personal Connections

Now that I have established that the project has been successful, I will now attempt to unpack the pleasure of *Bad News* by referring back to the ideas that were introduced in Chapter 3, particularly the *aesthetics of emergent narrative* that I identified there. A critical consideration here is that the aesthetics that are activated in this piece are rooted in features of the underlying simulation, which are in turn exposed to the player through a procedure of curation. As such, the pleasure of this experience is perhaps most fundamentally rooted in its utilization of curation, but I will save that discussion for Section 10.6.

First, analogies to both nonfiction (see Section 3.1.1) and lived experience (Section 3.1.2) obtain in this work, which activates the *aesthetics of the actual* and the *aesthetics of the personal*. As I have alluded to above, *Bad News* is about

Notizie, as I explained above.

[42]Fullerton said this in the course of a keynote presentation at the 2016 International Conference on Interactive Digital Storytelling, where Ben and I happened to be in the audience.

the splendor of everyday life. It is a game about death, but it does not handle death like games tend to, opting instead for a solemn treatment. While ostensibly there is a puzzle at the core of the experience, the notification task is really about examining a life that someone has led and is now leaving behind. As codesigner Adam Summerville noted in a filmed performance at the Slamdance Film Festival, "It's more about examining this person's life and figuring out who they were and who they affected and who is affected by their passing".[43] This grounding in everyday life (and everyday death) is further emphasized in the naturalistic conversation that undergirds the experience, as opposed to the stylized dialogue of fictional media. In total, *Bad News* has the flavor of nonfiction and, as such, the aesthetics of the actual are clearly at work at the level of its gameplay. Moreover, the underlying *Talk of the Town* character simulation—which we augment but do not contradict—both supports and reinforces these aesthetics.

An analogy to lived experience, then, obtains through the player's engagement with the apparently actual world of the town she visits. In his recent recollection of highlight moments, Ben Samuel cited the case of a player who deeply engaged the experience in an extended playthrough:

> I love it when the player recognizes the true weight of the responsibility they bear, and allow themselves to be swept up in the emotion of the final reveal. [...] One player, whose playthrough lasted for nearly two hours, made meaningful connections with essentially every person she met, culminating in an emotional final exchange where both player and next of kin were reduced to tears.[44]

Indeed, through interactions with the actor, whose improvisational skill and style of delivery maintain the verisimilitude, a player comes to actually *know* a set of *Talk of the Town* characters; by navigating one of its towns, she truly visits the

[43]Footage from this performance, including this quotation, is captured in a project spotlight video that was produced by Marie Jamora and Jason McLagan for Slamdance TV [536].

[44]Personal communication, June 14, 2018.

homes and businesses there.

I am claiming this by an appeal to reason, but the experience actually bakes in a mechanism by which this claim may be tested: the *epilogue*. Whereas a procedural town history (explored by livecoding with me) may intrigue an experiencer for being both synthetic and veridical—the aesthetics of the actual—the generated epilogue hits players on a more personal level. In both cases, the encounter is with raw *Talk of the Town* data displayed in a console window (see Figure 9.3), but in an epilogue the interactor has come to *know* the little chunks of data. One player, the researcher Mike Cook, was especially affected by the epilogue of his playthrough, and I wrote to him recently to ask why. Here is what he said:

> Bad News' epilogue has stayed with me to this day as one of the most impactful generative experiences I've ever had. Most simulation-based generative systems are impressive in a sort of Sublime-esque way, a feeling of awe derived from the scale of what is happening—that's the first hundred years of Bad News' simulation. The town is formed, businesses are founded, dynasties are formed, it's incredible to know it's happening but there's little at stake. But as I sat down to see the epilogue, in the brief flash as three more decades passed by, this wasn't a random jumble of names and events any more. Caught up in the whirlpool were people I had met, relationships I was invested in, unfinished stories that begged for an ending—it all felt very fragile, and terrifying. I've never been more invested in a generative system than I was at that moment.[45]

In another epilogue—the one following the playthrough that was featured in the last section—a player expressed deep remorse about something he had done in the town. To recapitulate, in this storyworld, a character had become a pariah after tanking a family restaurant that was a beloved town institution. When the player eventually met with this person, who had become a recluse, they engaged in deep conversation and eventually the player proposed a plan to repair the character's damaged reputation and life. While I think the player was sincere in

[45]Personal communication, June 20, 2018.

wanting to help the character, the plan was in fact a lie that pertained to the player's cover story—he was really attempting to gain information about the next of kin. In the epilogue, the player and I discovered that the character, Douglas Salva, had remained a recluse who, thirty years later, still worked as a stocker on a department store's night shift. Even though this was illustrated to the player through sanitary computer printout in a console window, Douglas Salva had become real to him, and he expressed a remarkably deep remorse at having caused this entity—*a mere bundle of data*—pain.

Inexhaustible Narrative Fields

I have already discussed at length (in Section 9.3) how *Talk of the Town* exemplifies what I have called the cornucopian approach to emergent narrative, through its *overgeneration* of emergent scenarios that charge each generated town with a considerable narrative potential. When a storyworld feels like an inexhaustible narrative field—as it may for a *Bad News* player who can go anywhere in a town and talk to any of its denizens, each of whom have lived out entire simulated lives—the *aesthetics of the vast* are at work. Furthermore, the *aesthetics of a larger context* are also yielded in that each *Talk of the Town* emergent scenario stands against a backdrop of other ones, and moreover the monotony of everyday existence that constitutes the far majority of a town's history. In *Bad News*, the deceased character works as a kind of narrative center of gravity, and so the emergent scenarios pertaining to that person's life are particularly imbued with connection to a larger context.

Shared Authorship

Distilling insights that we gleaned from discussions with players after gameplay, Ben Samuel writes in his own dissertation about how *Talk of the Town*'s simulation undergirds a sense of agency (citation inserted for clarity):

> many players expressed that the experience felt very unique—both by virtue of the fact that this is a gameplay experience unlike most others, but also because the player has free reign to explore a town with hundreds of characters in any way that they choose. Thus players report feeling high senses of agency over the shape of their gameplay session. Players have said that they felt transported to the world [420], and were able to readily visualize the people that they spoke with and the places that they visited. Many players found their towns so vibrant that they were shocked to learn during the postmortem that the towns were not designed by hand. [1090, p. 380]

In this way, and especially through her conversational interactions with the town denizens, the player *coauthors* an emergent story that is built atop the emergent narrative material generated by the underlying simulation. Another recent recollection by Ben provides a nice illustration:

> Many of my favorite memories hinge upon the addition of information provided by the player that complemented and augmented the baseline histories of the town's citizens provided by the simulation. At the 2016 CHI conference, a player struck up a conversation with a young man who painted houses, but had aspirations to move to a studio in New York and become a fully fledged artist. This player gave him very practical advice, and showed him how he could begin to pursue his artistic dreams without even needing to leave his hometown. Another player, when he first learned the name of the town, asked the mortician if he had heard about a recent hullabaloo involving a trolley car and some dogs. The mortician hadn't heard of it, and the player seemed disappointed. However, as the player walked around town, many of the folks he spoke with excitedly asked him if he had heard of the recent trolley trouble, to which he always enthusiastically said "yes!" As the player searched the town and found the next of kin, he slowly learned more and more about the story behind the trolley, none of

> which was actually modeled within the simulation, but that didn't contradict anything established by it either.[46]

As I have mentioned several times by now, Ben Samuel's dissertation presents his framework for *shared authorship*, which concerns experiences in which system and human collaborate to craft a narrative artifact that neither could have developed alone. In that context, Ben uses *Bad News* as the subject of a case study exploring these ideas [1084, pp. 384–405]—for the purposes of this dissertation, that writing may be consulted for an extensive explanation of how we cultivated an *aesthetics of the coauthored* in this project.

Finding the Story

A related phenomenon to shared authorship undergirds the *aesthetics of the uncovered*: when through endeavor one encounters an interesting story that would have otherwise gone unnoticed, a distinct pleasure obtains. In *Bad News*, stories are uncovered in a few different ways. Ultimately, they emerge from the simulation by virtue of some amount of story sifting, which means that I do a lot of the uncovering, since I am the wizard. When I extract potent material to pass off to Ben, it is extremely rewarding to hear it become realized, later on, in a conversational story. In these cases, the aesthetics of the uncovered are activated for me personally, which expresses the degree to which *Bad News*'s personnel are also consumers of the experience. Through discussion with him, I know that this pleasure also holds for Ben, even though he is the one delivering the conversational stories. In a recent paper [1091], collaborators and I wrote about the degree to which creators of experiences in the area of computationally assisted performance may also be consumers of them.

[46] Personal communication, June 14, 2018.

The aesthetics of the uncovered also prevails with regard to the experience of *Bad News* players. As I noted above, the storyworlds that drive this project are so vast that players can only plumb the surfaces of their narrative depths, which means that most of the latent interesting stories in a town go unnoticed by the player. When a player steers the conversation toward the revelation of a particular narrative, she uncovers something that would have otherwise remained hidden. This is especially true of cases in which player inquiries have lead me, as wizard, to extract narrative material that I had not already encountered.

Finally, in the case of our *augmented radio play* performances, audience members have frequently called out ideas that have guided my story sifting toward potent narrative outcomes. When this happens, the live audience directly experiences the aesthetics of the uncovered.

Small Miracles

I have already explained, in Section 9.3, how many of *Talk of the Town*'s evocative emergent scenarios result from simple probabilistic procedures. Many of these, such as the restaurant that operated for over a century, seem to be intriguing simply because they are improbable. As Ben Samuel explains in his recollection of the playthrough involving that restaurant, in *Bad News* a kind of higher-order improbability may also obtain according to unlikely connections between the player's improvisation and the underlying simulation:

> One of my favorite recurring themes is the serendipity that transpires when information concocted by the player dovetails so perfectly with the specifics of the simulation that it feels like it must have been planned and scripted from the start. One such example occurred during a playthrough that took place during San Francisco's 2016 "Come Out And Play" Festival. The player of this particular run had conceived of an ingenious cover story, that no previous player had ever presented: they were a restaurateur hoping to open a dinner theatre

> in town, searching to cast someone to play a character that matched the description of the deceased to a T. [...] Eventually the player was directed towards someone who would be able to help him, but who had some tragic family history of his own. This would-be informant came from a long line of restaurateurs himself—his great-great-uncle (for some number of greats) established a restaurant over a hundred years prior. This restaurant had been a staple of the town, and was passed down from father to son for many generations, until finally it was bequeathed to an owner who would never have any children of his own. When this owner chose to retire, he passed the restaurant on to his nephew, who had been a successful server at this restaurant all his life. However, less than a year later, the nephew managed to drive the restaurant into the ground; a century's worth of family history and town pride forever lost. [...] A town that has a restaurant that lasts nearly a hundred years is a near impossible occurrence in Bad News. Players who utilized the specifics of their cover story to create an emotional connection with the denizens of the town were almost just as rare. For both to transpire in a single playthrough, and to be so deeply interwoven into each other, was a small miracle, and will forever remain one of my favorite Bad Newsian experiences.[47]

Intrigue rooted in improbability is what yields the *aesthetics of the improbable*, and when player improvisation serendipitously connects to remarkable underlying material, the aesthetics are doubly activated.

Embracing the Uncanny

In *Bad News*, we consciously design toward the *aesthetics of the uncanny* by acknowledging or even emphasizing *Talk of the Town*'s peculiarities at the surface of gameplay. Earlier, in Section 9.2.5, I discussed a bug in the engine's social simulation that produces romantic connections between characters with massive age differences. Like I mentioned in that section, through playtesting we discovered that this uncanny element actually yielded considerable intrigue—players were compelled to inquire about these bizarre relationships—and so we kept the 'bug'

[47] Personal communication, June 14, 2018.

and embraced the uncanny.

As another example, in one of my favorite *Bad News* moments, Noah Wardrip-Fruin visited the home of a next of kin only to find that a four-year old child was home alone there. This curious situation was produced by the *Talk of the Town* subsystem that handles character routines, outlined in Section 9.2.4. Specifically, when a parent goes out on an errand, the routine system neglects to model that any young children who are at home should probably be brought along on the trip. In this case, the child's mother had gone off to a jewelry store, but Noah's player interface expressed that someone was inside the house making a small noise. He knocked, but no one answered—this is because a procedure for deciding who comes to the door specifies that small children never do. Noah then spoke aloud that he wished to look through the window, and so I extemporaneously typed a description of a small child watching television and displayed it on the player interface. Next, Noah did something that remarkably no other player has ever done—he broke into the house. This forced Ben to play the role of a four-year-old boy, which he did masterfully, and through considerable effort Noah learned from the boy that the mother had gone to the jewelry store.[48] This scene was bizarre in a good way—the aesthetics of the uncanny prevailed.

Disposable Worlds

Finally, the *aesthetics of the ephemeral*: no player's town has been visited before, and no one will go there again. To conclude the epilogue, I terminate the *Talk of the Town* simulation instance and the player and I watch as her little

[48] I have already mentioned that, in Tarn Adams's playthrough, Ben was tasked with playing a 111-year-old woman who was the next of kin. Considered together, these roles illustrate the breadth of characters that Ben may have to play at any moment. That he does this so well is impressive, especially considering that he does not know his next role until moments before a scene, and moreover none of the characters that he plays even exist until the hour of performance. The kid has chops!

world disappears back to nothing—it was only an electric wind.

10.5 The Pain of *Bad News*

Owing to the success of this project, there is no need for me to take on the third person, as I did for the brutal critique of *Diol/Diel/Dial* that was provided in Section 8.4. While a lack of curation was identified as a fatal flaw of that experience, *Bad News* is a full-fledged curationist work that instantiates the entire pipeline of the curationist architecture. This allows raw storyworld material to ultimately be expressed in constructed narrative artifacts that are mounted in the media experience, in this case through conversational storytelling.

Toward Automatic Curation

Of course, curation in this project is human-powered, and this could be viewed as inflicting a distinct kind of pain. As I have expressed at several points in this document, the ultimate vision of curationist emergent narrative is one of fully automatic curation, but while this work is curationist, it does not come very close to meeting that vision. Indeed, in his own dissertation, my collaborator Ben Samuel also identifies human reliance as a potential weakness of this project:

> The most damning criticism of Bad News in a dissertation about creating computational works that enable shared authorship is that it relies significantly on humans to achieve its goals. The presence of humans affords wonderful advantages, [...] but sadly without humans at play there are no computational mechanisms to fall back on. [1084, p. 400]

In fact, Ben even proceeds to pinpoint a lack of automatic curation of simulated material, like I am doing in this section:

> For example, I have discussed how all of the narrative in the system is emergent. The actor and the wizard are capable of leveraging

> their natural human proclivity for recognizing narrative to unearth the narratively interesting sequences of events from the town's entire simulated history. That is to say, though these interesting sequences were created by the simulation, the simulation itself has no recognition of their significance. [1084, p. 400]

To be clear, I do think that computationally assisted live performance—specifically, hybrid works that incorporate human performers—is an exciting nascent area that we should not seek to computationalize. I know that Ben agrees with this too. What I am attempting to articulate here is that the prospect of fully automatic curation is itself alluring, in its own way, and in this regard *Bad News* tantalizes with its suggestion of how that might feel, because though it incorporates computation, its curation is in no way automatic.

The Cover-Up

Another ramification of the heavy utilization of humans in this project is related to a distinct appeal of that morphology: when humans are in the loop, the failings of underlying computational systems can be covered up quite easily. Much of the pain of *Diol/Diel/Dial* was constituted in blunders at the level of *World*'s simulation, as I expressed in Section 8.4, but in *Bad News* the errors of *Talk of the Town* do not really make it to the surface. Where the simulation lacks, the wizard and especially the actor augment, filling in its modeling gaps and injecting causal links that are not there and so forth.

As I have stated several times now, I believe that computational techniques that are intended to support media experiences cannot be truly appraised except through actual implemented experiences of the kind that they are meant to support. This is not an original idea, but rather it follows earlier articulations by several other researchers [762, 1200, 505, 610, 1316]. As an expression of this sentiment in this document, I do not appraise my simulation engines outside of the case studies that dissect actual media experiences built atop them. In the case of this case, however, we have a media experience that, through human power, is adept at leveraging a simulation engine's strong points while simultaneously hiding its weaknesses (and even potentially its errors).

As such, this makes genuine appraisal of the underlying simulation difficult, but incidentally such an evaluation will in fact appear in this document, though elsewhere. In the story that opens the next chapter, I discuss an exploratory project in which I attempted to implement automatic curation of *Talk of the Town*'s storyworlds. My specific aim in this effort was to automatically sift for interesting emergent scenarios, with the idea that such material could then be passed off to a narrativizer that I would build in turn. Essentially, this would mean loosely operationalizing both the wizard and actor strategies that are employed in *Bad News*. What I found, however, was that this was essentially impossible due to particular deficits in the *Talk of the Town* simulation. As such, in telling this story, I will ultimately provide a more detailed appraisal of that engine.

10.6 Curationism and *Bad News*

In this project, the wizard and actor collaborate to construct stories out of the emergent material of a *Talk of the Town* storyworld, which are then mounted into the *Bad News* experience through the modality of conversational storytelling. As I have already shown in discussing the extensive example of curation in Section 10.3, this instantiates the curationist architecture outlined in Section 5.3—specifically the *feedforward curation* variant—though in a way that is largely dependent on human power. As such, *Bad News* represents an intermediate stage in my curationist practice, one that spans between *Diol/Diel/Dial*'s lack of curation and the fully automatic curation of *Sheldon County*. In this section, I will briefly summarize how *Bad News* instantiates the framework for curationist emergent narrative that I presented in Chapter 5.

As I did in my earlier case study of *Diol/Diel/Dial* (see Section 8.5), I will now proceed to answer the following question: for each curationist architectural component, who or what in this project acts as that component? Figure 10.9 illustrates these concerns with regard to *Bad News*, which I will also outline in prose, component by component:

- **Experiencer.** *Bad News*'s *experiencer* is the player who sits in the gameplay area of our physical installation and explores the generated storyworld to speak with its residents. As I have noted, in special cases we allow for multiple players, in which case each one is an experiencer. In some special cases, there may be other experiencers who are not players. For example, in our so-called *augmented radio play* variant, described in Section 10.2.5, audience members may also experience the piece, but not in the same way that the player does. For the sake of simplicity and comprehensibility, Figure 10.9 does not capture these considerations.

- **Simulated storyworld.** In each *Bad News* performance, a *Talk of the Town* simulation instance is the *simulated storyworld*. It is the procedurally generated American small town that the player explores to find and notify the next of kin of one of its residents who has died.

- **Chronicler.** As with *Diol/Diel/Dial*, there is no distinct *chronicler* module at work in *Bad News*, but rather the associated functionality is built into the *Talk of the Town* code. To simplify, we can say that *Talk of the Town* is its own chronicler.

- **Chronicle.** A *chronicle* in this project comprises all of the various data that

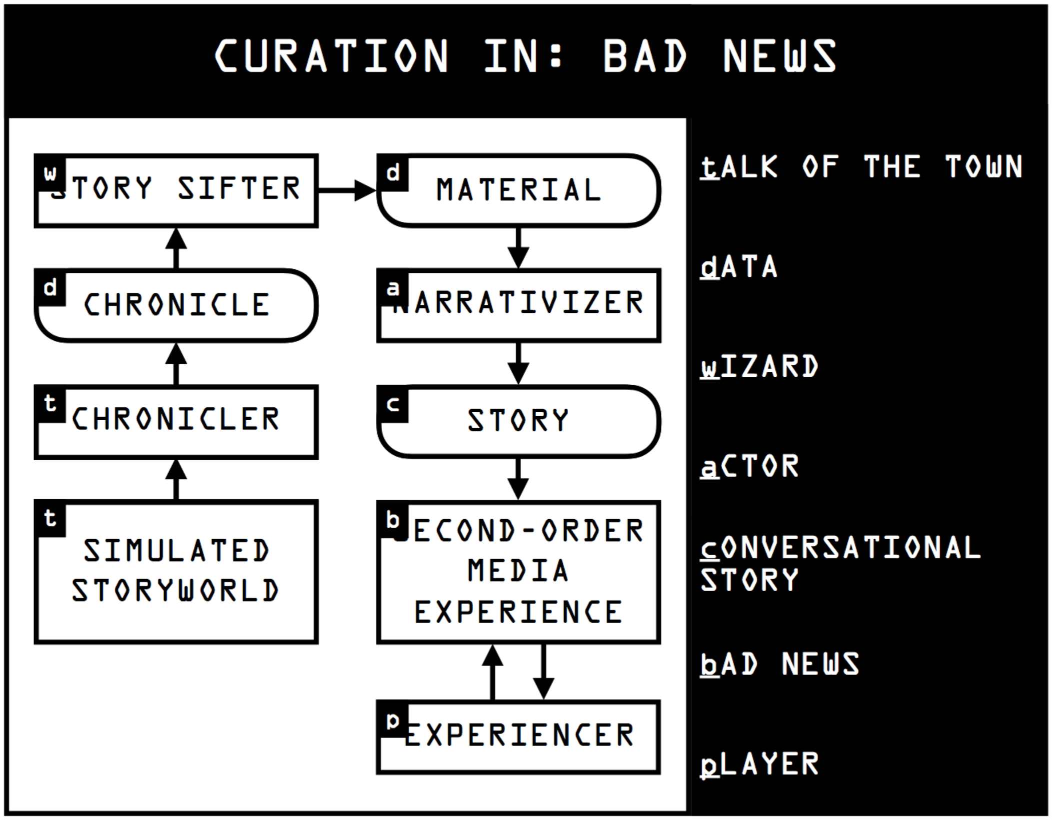


Figure 10.9: Curationist emergent narrative in *Bad News*. In this diagram, letter designators placed in the corner of each architectural component cue the corresponding entities in the *Bad News* project. Here, we find a variation on the *feedforward curation* pattern shown above in Figure 5.2: the transpiring of a *Talk of the Town* storyworld is captured in a chronicle, which a human wizard sifts through to deliver narratively potent material to a human actor, who narrativizes that material to build narrative artifacts that are then mounted into the *Bad News* experience through conversational storytelling. As a successful proof of concept of curationist emergent narrative, *Bad News* begs the question: what would such a project look like if this kind of curation was carried out without human help?

is queryable in the wizard console, which is a Python interpreter wrapped around an active simulation instance that is stored in working memory. This is all of the data that is tracked by the *Talk of the Town* simulation engine, and it represents a stored record of much of what has transpired in the history of the town at hand.

- **Story sifter.** Unlike *Diol/Diel/Dial*, this project actually employs a *story sifter.* As I explained in great detail in Section 10.3, in *Bad News* the human wizard does story sifting. Specifically, I do livecoding to sift through the accumulated data representing the history of the town, my aim being to uncover interesting emergent scenarios (of the kind outlined in Section 9.3), especially ones that connect to the deceased character in some way. It could also make sense to say that the actor interface also sifts this material, since it dynamically updates (as the experience proceeds) to display storyworld information to the actor. However, the data displayed on this interface tends to lack narrative intrigue; rather than the makings of storylines, it expresses the basic material that Ben needs to perform. As such, I think it is probably not worth considering this notion in detail, and Figure 10.9 does not capture this interpretation.

- **Material.** In this project, the sifted *material* is textual data that the wizard delivers to the actor via a chat interface (the actor–wizard web chat). Figure 10.3 shows an example of such marerial, as seen in a screenshot of the chat taken during a performance.

- **Narrativizer.** Again *Bad News* prevails over *Diol/Diel/Dial*, in this case because the experience itself, rather than a human interactor, carries out narrativization. As I discussed at length in Section 4.2.1, a fundamental pitfall in emergent narrative is simulations that do not understand their own storyworlds and instead offload the burden of narrativization onto their human experiencers. In the case of this project, the simulation does not understand its own storyworlds, but a human duo works to curate them so that actual narrative artifacts may be constructed and mounted in the experience. In this configuration, the actor is the *narrativizer.* As I showed in detail in Section 10.3, one of Ben's major tasks as actor is to narrativize the excavated material that I deliver to him to build narrative artifacts that are mounted into the *Bad News* experience through conversational storytelling.

- **Story.** In *Bad News*, a *story* is told by the actor in the course of a conversation with the player. In the form of performance transcripts included in Section 10.3, I provided several examples of such stories. Like I have noted several times above, a constructed story may be delivered in any modality that supports narrative. For example, in Section 3.1.1, I provided a case study of curationist emergent narrative that centered on Tim Denee's *Oilfurnace*, which is a comic that recounts the emergent intrigue of a *Dwarf Fortress* storyworld. Clearly, curationism utilizes an ecumenical notion of 'story', and these two examples express the wide range of modalities that may be targeted in works following this framework. With *Sheldon County*, I am again targeting the modality of *oral narrative* [90], but in that case the delivery is monologic instead of conversational.

- **Media experience.** In curationism, the term 'media experience' specifically denotes the experience that curated narrative artifacts (each being a 'story') are mounted into for human encounter—thus, in this project, the *media experience* is *Bad News* itself. Because this experience is not the simulated storyworld, but rather a decoupled installation-based work, the project uses the architectural variant of *feedforward curation*. By this variant, described in Section 5.3.2, stories that are constructed out of the generated material of a storyworld are mounted into a decoupled *second-order* experience. I realize that the matter may be a bit confusing in this case, since a *Bad News* player *does* explore the simulated storyworld recounted by the stories that are imparted to her. The distinction here is subtle and it arises from the peculiar human-powered nature of the project. In *feedback curation*, constructed stories are fed back into the storyworld itself, since that is the mounting target. In *Bad News*, curation is not cybernetic because the stories that Ben constructs are not actually fed back into the storyworld—when he delivers a story to the player, I have no means with which to actually inject that story into the world through the mechanism of *Talk of the Town* simulation. This is there is no representational scheme by which I could encode a conversational story in a way that the system would actually be able to process. In feedback curation, the story that is fed back into the storyworld is actually understood by the simulation that models that world. An illustrative case of this would be a videogame where non-player characters tell stories to one another about what has happened in the world, and critically in such a way that the recipients of those stories actually process them in a meaningful way.[49]

Summary

To recapitulate, *Bad News* is a full-fledged work of curationist emergent narrative, but one that relies heavily on human power. Specifically, this project that

[49]While I am not aware of any examples of such a thing—though it was the basis for my original dissertation topic and I would like to return to the idea at some point—the example of feedback curation that I gave in Section 5.3.2 was *Eve Online* player-constructed propaganda—about which Marcus Carter has written [166]—being broadcasted in-game (thus, in the actual storyworld). In this case, the simulation does not understand the stories, but the human players do, and they may respond by taking action that actually changes the world. In the case of *Bad News*, Ben may play a character who tells a story and is affected by the telling, but the experience has no means by which the underlying representation of that character, in *Talk of the Town*, may be altered accordingly. As such, the *Talk of the Town* storyworld does not change when stories are constructed out of its generated material. Generally, while *Bad News* takes place in those storyworlds, at a technical and ontological level, it more precisely takes place in a kind of liminal projection of that world onto the high-fidelity space of our world (and onto the modality of face-to-face conversation). It would be interesting to explore how Ben's curated stories might somehow be injected back into the storyworld, perhaps after gameplay. While the epilogue does not incorporate anything that happens during a performance, as I explained in Section 10.2.4, I really wished it did.

employs the *feedforward curation* variant of the curationist architecture (see Section 5.3.2), since it mounts stories curated out of *Talk of the Town* material into a second-order experience, *Bad News*. In its instantiation of this pattern, a human wizard works a story sifter to deliver narratively potent raw material to a human actor, who narrativizes that material to build narrative artifacts that are then mounted in the experience through conversational storytelling. As a successful proof of concept of curationist emergent narrative, *Bad News* begs the question: what would such a project look like if this kind of curation was carried out without human help? Later, in Chapter 12, I outline my ongoing exploration of fully automatic curation in the generative podcast *Sheldon County*.

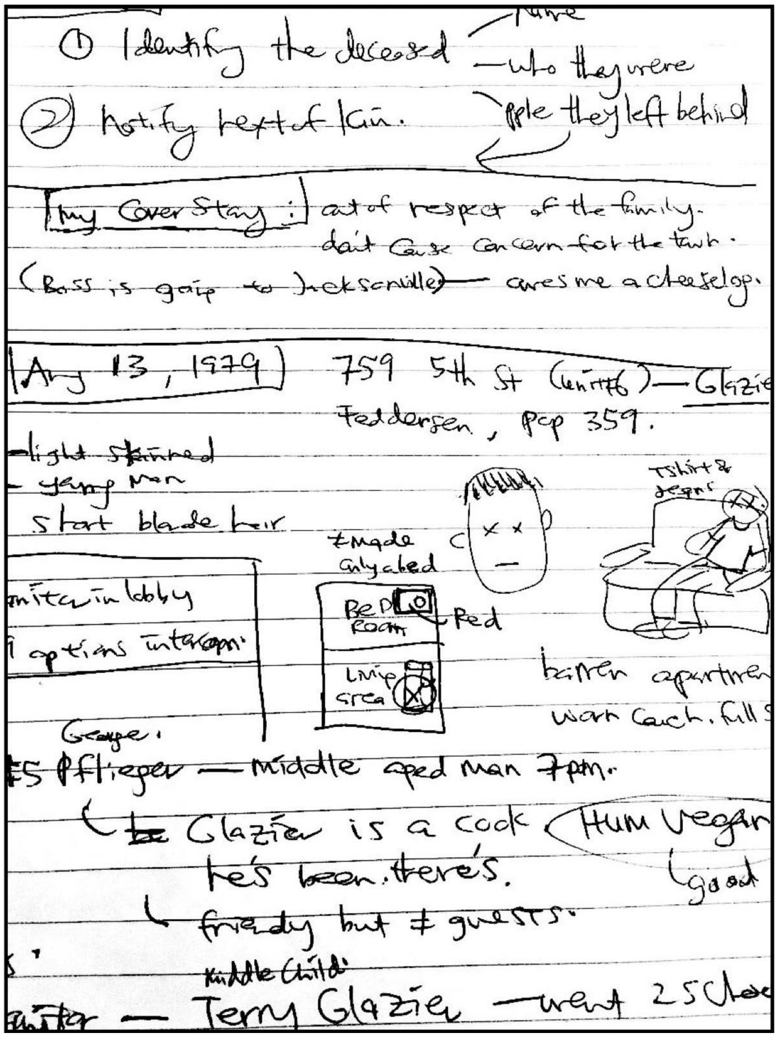


Figure 10.10: A *Bad News* player's notes map out various emergent concerns surrounding the life of a late *Talk of the Town* character. As the scrawl expresses, Terry Glazier was a young man who worked as a cook at a restaurant called Hum Vegan, in the town of Fedderson (pop. 359). This effort to make sense of such emergent material is an example of *mental curation*, the impoverished curationist architecture that I identified in Section 5.3.2. Indeed, these notes may be understood as a kind of narrative artifact that is curated out of the emergent material of the *Talk of the Town* storyworld of Fedderson. In this project, however, the experience itself carries out a full-fledged procedure of *feedforward curation*, whereby *Talk of the Town*'s emergent material is curated and then mounted into the *Bad News* experience through conversational storytelling. As such, it is a true example of curationist emergent narrative, though one that is in large part human-powered.

Chapter 11

Hennepin

The construction of the system is inevitable.

Sheldon Klein

Toward the end of my development work on *Talk of the Town*, I became interested in the idea of automatically identifying the kinds of narratively potent emergent scenarios that undergird each *Bad News* performance. This would mean *automatic story sifting*, as opposed to the manual procedure that I carry out in my capacity as wizard in that piece. Since my method for doing this myself relied on livecoding queries to a simulation instance, it seemed like I could simply have the automatic story sifter utilize the same code sequences that I construct on the fly during a performance. Of course, my sifting process is generally exploratory and improvisational, as the examples in Section 10.3 illustrated, but I figured that I could start by crafting functions for identifying recurring scenario types that I frequently excavate, such as cases of *unrequited love* or *love triangles*. Later on, I could explore the development of *sifting heuristics* that the automatic story sifter could use to search for material that is intriguing or pleasurable—this could work something like Doug Lenat's system *AM* [674], which carries out heuristic exploration of a possibility space to automatically discover interesting or aesthetically pleasing mathematical concepts.

As a first move in my investigation of automatic story sifting, I wrote up little functions that could be used to search over a town's history to identify particular scenarios that tend to emerge in *Talk of the Town* towns. For example, here is some Python code that can be used to automatically excavate cases of unrequited love (and package them up as objects of an `UnrequitedLove` class, whose definition is not included here):

```
def excavate_unrequited_love_cases(town):
    """Sift for cases of unrequited love."""
    unrequited_love_cases = []
    for lover in town.residents:
        for loved in lover.loves:
            if lover not in loved.loves:
                unrequited_love_cases.append(
                    UnrequitedLove(
                        lover=lover,
                        loved=loved
                    )
                )
    return unrequited_love_cases
```

In this code, the attribute `loves` resolves to a list containing all the other characters with whom one is in love, where being in love is operationalized as holding a spark value above a certain threshold. As the listing expresses, the function would not just excavate a single case of unrequited love, but every single case in the town's history. Here is another function that sifts through this accumulated history for cases of a more specific phenomenon, the love triangle:

```
def excavate_love_triangles(town):
    """Sift for love triangles."""
    love_triangles = []
    for a in town.residents:
        for b in a.loves:
```

```
        for c in b.loves:
            if a in c.loves:
                if a not in b.loves:
                    if b not in c.loves:
                        if c not in a.loves:
                            love_triangle.append(
                                LoveTriangle(
                                    a=a,
                                    b=b,
                                    c=c
                                )
                            )
    return love_triangles
```

This function finds all cases of character triads $\{a, b, c\}$, such that a is in love with b, b is in love with c, c is in love with a, and none of the love is reciprocated.

This was fairly interesting, and it was pretty easy to craft functions like these, but much of the intrigue in *Talk of the Town*, and in turn in *Bad News*, concerns the *stacking* and *interlocking* of multiple emergent scenarios. For example, let us reconsider Adam Summerville's favorite scenario to emerge in the simulation, which I quoted in Section 9.3:

> My favorite situation to arise from the simulation was something like a Shakespearean tragedy, set in small town America. The deceased was the scion of a family that had run one of the two bars in the town, with a history nearly as long of that of the town itself, and he was in love with the daughter of the owner of the other bar in town. The two bar-owning families had hated each other for generations, but the two children of the families were able to get past that and fall in love. For this to happen, only to have tragedy befall their star-crossed love feels like Bad News was cribbing from one of my favorite movies, "Scotland, PA"—a retelling of Macbeth in a fast food restaurant in the 1970's. I will always be amazed by the kinds of emergent stories that Bad News is capable of generating.[1]

How would the automatic story sifter excavate this material? My sense was that a good approach would be to break down such a situation into its component sce-

[1] Personal communication, June 14, 2018.

narios and then craft functions so that each of those scenarios could be recognized. From here, the next step would be to specify how instances of the component scenarios interlock to produce the targeted higher-order structure. If it were equipped with a capacity to both sift for the components and reason about the interlocking of instances of each, the automatic story sifter could excavate examples like Adam's by finding cases that satisfy the interlocking criteria.

Let us work through this in a detailed example. First, what are the components of Adam's example? At its foundation is a bitter rivalry between two long-established family businesses, which could be decomposed into a more specific explanation such as this: there are two businesses that have existed in the town for a long time, the businesses are family-owned, the businesses are in competition, and the family members that run the businesses do not like one another. To start, here is a function for sifting out long-running businesses:

```
def excavate_town_institutions(town):
    """Sift for cases of long-running businesses."""
     # Operationalize 'long-running' by number of years
    threshold = 40
    town_institutions = []
    for business in town.businesses:
        if town.year - business.founded >= threshold:
            town_institutions.append(
                TownInstitution(
                    business=business
                )
            )
    return town_institutions
```

This allows the sifter to excavate long-running businesses, but it also needs to know which ones are family businesses. One way to operationalize this is to say that a family business is one whose owner was preceded by an owner of the same family. The following code captures this:

```python
def is_family_business(business):
    """Return whether the given business is
       family-owned."""
    owner = business.owner.person
    previous_owner = business.owner.preceded_by.person
    if previous_owner in owner.extended_family:
        return True
    return False
```

Of course, this notion could have been operationalized differently. We could instead only accept circumstances in which every owner of the business was related to the previous one, or we could say that the business must be passed from parent to child each time, or what have you. In my view, it is probably best to equip the automatic story sifter with liberal policies, since that increases the chance of excavating usable material. As for the notion of two businesses being in competition, this could be operationalized by them being of the same type (e.g., `bar`), but a richer (and also more inclusive) policy might instead look at whether the businesses have services in common:

```python
def competitors(business, other_business):
    """Return whether the given businesses are in
       competition."""
    business_services = set(business.services)
    other_services = set(other_business.services)
    if business_services & other_services:
        # Set overlap, thus common service(s)
        return True
    return False
```

Finally, the sifter requires means by which animosity between the families may be detected. While a strict notion might require something like extreme disdain between each and every member of the respective families, a liberal (but still reasonable) policy might simply require animosity between the respective business owners. Here is a function that captures the latter by operationalizing animosity using *Talk of the Town*'s existing `enmity` relation:

```
def animosity(character, other):
    """Return whether there is mutual animosity between
       the given characters."""
    if character in other.enemies:
        if other in character.enemies:
            return True
    return False
```

Now, to sift for bitter rivalries between family businesses, which is the foundation of Adam's example, the sifter can simply search for cases of interlocking between each of the component scenarios. Given the above code examples, it may do this automatically by using a function such as this:

```
def excavate_family_business_rivalries(town):
    """Sift for cases of bitter rivalries between
       long-running family businesses."""
    family_business_rivalries = []
    town_institutions = excavate_town_institutions(town)
    for business in town_institutions:
        if not is_family_business(business):
            continue
        for competitor in town_institutions:
            if not is_family_business(competitor):
                continue
            if not competitors(business, competitor):
                continue
            owner = business.owner.person
            other_owner = competitor.owner.person
            if animosity(owner, other_owner):
                family_business_rivalries.append(
                    FamilyBusinessRivalry(
                        rivals=(business, competitor)
                    )
                )
    return family_business_rivalries
```

Finally, at the heart of Adam's scenario is a case of forbidden love between members of the rival families. In his specific example, one is the owner of one of the businesses and the other is the daughter of the rival owner. While we could certainly detect such specifics, it is again probably a better idea to be more indiscriminate, so as to increase the likelihood of successful sifting. Here is code that the automatic story sifter could use to excavate forbidden love of the type that is central to Adam's example:

```
def excavate_forbidden_love(town):
    """Sift for cases of forbidden love between
       members of hostile families that operate
       long-running rival businesses."""
    forbidden_love_cases = []
    family_business_rivalries = (
        excavate_family_business_rivalries(town)
    )
    for rivalry in family_business_rivalries:
        business, competitor = rivalry.rivals
        first_family = (
            business.owner.person.immediate_family
        )
        second_family = (
            competitor.owner.person.immediate_family
        )
        for lover in first_family:
            for loved in lover.loves:
                if loved in second_family:
                    if lover in loved.loves:
                        forbidden_love_cases.append(
                            ForbiddenLove(
                                lovers=(lover, loved),
                                rivalry=rivalry
                            )
                        )
    return forbidden_love_cases
```

In this code, only members of the rival business owners' *immediate* families are eligible for forbidden love, because here too liberal a definition (such as extended family) could potentially lead to the excavation of false positives, such as lovers who are actually related to both families (through in-laws or distant cousins). In any event, defining such *sifting patterns* (see Section 5.4) requires iteration and testing to make sure that the patterns are liberal enough to lead to the successful excavation of material in most storyworlds, but not so indiscriminate as to admit cases that do not actually instantiate the targeted scenario.

The ultimate goal of the story sifter is to excavate narratively potent material, and in the above code examples, such extracted material is packaged into objects of scenario-specific classes, such as `ForbiddenLove`. Rather than package up all the material that may be relevant to a scenario, the sifter includes only the critical components that underpin it—for example, the two lovers and family-business rivalry that inhibits their love. As the code examples also illustrate, such packages of material—let us call them *nuggets*—may be referenced in the sifting patterns that are used to sift out other material. As such, nuggets may be used to compose the material of other nuggets, such as with the `ForbiddenLove` class's `rivalry` keyword argument (which expects an object of the class `FamilyBusinessRivalry`).[2] While not all of the relevant material is packaged up, the critical components can work as hooks into the chronicle that may be utilized to extract other additional material. For example, the components of a `ForbiddenLove` nugget can work as hooks by which the owners of the rival businesses and the relationships between those owners and the lovers could in turn be extracted.

In the fall of 2016, I developed an automatic story sifter that worked according to the principals that I have just outlined, but it led me to a painful realization

[2]Here, I am using Python terminology simply because the above code examples are in that programming language. Of course, these are basic concepts and implementations may vary wildly—indeed, any programming language or computational framework could be utilized. With these examples, my goal is to elucidate story sifting at some level of detail, so that is why I use code listings and terminology.

about *Talk of the Town*: the simulation's representation of character actions is too abstract for actual emergent stories to obtain. The system could extract intriguing nuggets, such as cases of forbidden love, but the point of story sifting is to hand off the excavated material to a *narrativizer* module that actually constructs a narrative artifact out of that material. As abstract scenarios, the extracted *Talk of the Town* nuggets made narrativization an extremely difficult challenge.

For example, if a narrativizer was to tell a story of forbidden love, given a nugget capturing such a scenario, how would it do it? It could inspect the chronicle for data capturing the history of each business, but there would not be much intrigue there: the history of a business in *Talk of the Town* is essentially a series of names and dates capturing who owned it and who worked there and for how long. Certainly the narrativizer would want to tell the story of how the lovers fell in love and how the owners became enemies, but the data capturing such concerns is a series of abstract social interactions:

- `January 13, 1973: social interaction at Zilencio [lovers meet]`
- `Feburary 22, 1973: social interaction at Zilencio [attraction grows]`
- `April 6, 1973: social interaction at Horswill Farm Park [attraction grows]`
- `October 9, 1973: social interaction at Lessard's Lounge [attraction grows]`
- `...`
- `March 11, 1975: social interaction at Zilencio [lovers fall in love]`
- `August 21, 1975: social interaction at Zilencio [lovers fall more in love]`
- `...`

There is some minor intrigue here—for instance, Zilencio being an important place for the lovers—and the system could exploit that, but to tell an actual story that recounts such abstract concerns would require the narrativizer to essentially carry out all of the invention that is required in top-down story generation. Of course, this is exactly the narrativization task that Ben carries out in his capacity as *Bad News* actor. As I showed in Section 10.3, in our curation set-up I manually excavate narratively potent nuggets and deliver these to him via chat, and he then augments them to build stories that he imparts to the player conversationally. For *Bad News*, this abstractness is actually an appeal, since Ben would not have time to internalize a detailed emergent story, and moreover it would difficult to avoid contradicting one because that would require the memorization of myriad details. So while operationalizing Ben's narrativization process would make for a very interesting project, it would be an especially daunting one due to the abstractness of *Talk of the Town*'s generated material.[3]

As I was exploring the prospect of narrativizing *Talk of the Town*'s abstract material, I recalled a critique that I had received in an anonymous review of a paper on the system's social simulation [1043]. Here is an excerpt from Reviewer #2's feedback on my original submission:

> The relationship between two characters is modelled by two numbers: platonic affinity and romantic affinity. In such a simple two-factor model, most of what makes relationships interesting has been lost. [...] the *reason* for the relationship to be in the state it is has been lost. E.g. Smith hates Jones *because* he ripped him off. [...] it just updates the relationship deterministically based on the stats of the two characters. It would be impossible in this model for X to remember the reason why he hates Y because there was no action to begin with

[3] One prospect for a *Talk of the Town* automatic story sifter is to serve as a *Bad News* wizard's assistant: the system could excavate nuggets for use by the wizard, to save on the time and energy of excavation by livecoding. Such a system would be especially useful for wizards who are not me (and thus not as familiar with the *Talk of the Town* codebase), such as Jonathan Lessard (working in his Montreal troupe with actor Noah Drew) and anyone who would play the 'home version' of the game that we have discussed [1084, p. 383]. As I noted above, Adam Summerville is interested in developing the latter as one of his first research projects at Cal Poly Pamona, where he is now a professor. If he does find the time and energy to develop the project, I imagine an automatic story sifter will be a central component of the framework.

that prompted this hatred.[4]

The reviewer had pinpointed the very problem that I was encountering in my attempt to narrativize the material excavated by my automatic story sifter.[5] Moreover, while the critique captured that character relationships are too abstract, the more troubling issue is that *all character activity* in *Talk of the Town* is too abstract for narrativization. The only fine-grained actions that characters take are social interactions (abstractly construed), going to a place (as part of the routine modeling), being at work, being at home, being on an errand, and a few others.

Of course, there are a series of major life events that are modeled—marriage, divorce, birth, death, business founding, home construction, and a few more—but these are also fairly abstract. For example, when a house is constructed, there is no duration to that action, but rather it just suddenly transpires all at once. It may involve some component actions, like contracting an architect and builders who are working at the construction firm that day being recorded as having worked on the job, but still this is fairly abstract. More troublingly still, none of these events are particularly interesting—rather than providing narrative intrigue, they are meant to drive the evolution of the town's infrastructure. Finally, the only activity that is modeled at high granularity is the mechanics underpinning the simulation of character knowledge phenomena, but due to the nature of what

[4]Anonymous personal communication, June 6, 2016.

[5]To be fair, as I emphasize in that paper [1043], *Talk of the Town*'s social simulation was designed to be extremely lightweight in order to save memory and computation for the heavyweight modeling of character knowledge phenomena that is the hallmark of the system. In the case of *Bad News*, we need the system to create the next town in a few minutes or less, and a richer social simulation would likely preclude that, considering the number of characters in the storyworld. Indeed, as I will explain shortly, in *Hennepin* it may take many hours for a storyworld's history to be simulated, but I am specifically targeting second-order experiences that can work under that constraint. For example, in *Sheldon County*, the subject of the next chapter, an entire non-interactive media experience (a generated podcast) is produced all at once following the creation of a *Hennepin* storyworld. This introduces a latency on the order of a day between a potential experiencer deciding to experience the work and her actual engagement of it, but this is not a major problem given that the entire experience is produced at once, in full.

beliefs are formed about—largely benign appearance features, such as the size of a character's nose—little intrigue is yielded by this subsystem either.

In total, while I was thrilled with the shape that *Bad News* had taken, I had become disenchanted with the prospect of using *Talk of the Town* as the simulation engine in a fully automatic work of curationist emergent narrative, which is what I wanted to build next. As such, I decided to develop a new simulation engine, and I wanted it to work like a combination between *Talk of the Town* and *Comme il Faut* [804], which is the AI system that undergirds *Prom Week* [799, 803, 1085]. In *Comme il Faut*, characters reason over the state of the storyworld to decide which *fine-grained actions* they want to take on a given timestep; the actions authored for *Prom Week* include examples such as `flirt`, `insult`, and `ask out`. As such, my idea was to build a new engine that would again model the history of a storyworld, since that is a hallmark of my practice, but in that period characters would now take fine-grained actions. So rather than, for example, two characters Michael and Noah having an abstract social interaction on the night of October 5th, 1982, at the Law Offices of Mateas and Wardrip-Fruin, the two might `reminisce` at that time and place. Critically, these fine-grained actions would evolve relationships, so that a case of forbidden love, to revisit the example from above, would emerge out of a history of concrete character interactions. Likewise, I wanted actions to also serve as the primary drivers of the evolution of the storyworld itself—this would enable the generation of concrete action sequences that could be sifted to support feasible automatic narrativization into recounted emergent stories.

My initial plan for doing this work involved the integration of *Talk of the Town* and *Ensemble* [1089], which is a successor to *Comme il Faut* that democratizes its technology so that *Prom Week*-style 'social physics' may be utilized in other projects. Here is how I outlined this plan in my original thesis proposal (which featured a temporary renaming of *Talk of the Town* to *Talk/Town*):

> We are now working to integrate the Ensemble Engine with *Talk/-Town*, which will enable us to use its authoring tool to quickly specify rules for agent decision making. In an extension to the system that we are currently developing, agents will also use these decision procedures to select finer-grained social actions to take (*e.g.*, *flirt*, *insult*, *confide*, *reminisce*) on simulated timesteps. [1038, p. 11]

While eventually I decided to develop my own approach to modeling character actions, the methodology of the *Comme il Faut* and *Ensemble* tradition remained a major influence. For those that are familiar with both projects, I think '*Talk of the Town* meets *Prom Week*' is a good descriptor for the simulation engine that would eventually culminate from this refined approach.[6]

Hennepin is a simulation engine that models an American county over the course of its history, with particular attention to the lives of the characters who live there. It departs from my earlier simulation engines, *World* and *Talk of the Town*, by modeling its storyworlds with much more detail. Instead of abstract social interactions and monolithic life events, its generated histories are constituted in accumulations of concrete character actions that are modeled at roughly the level of granularity of *Prom Week*'s social actions. Due to the way that action selection works—this is explained in Section 11.2.7—character actions may contingently unlock subsequent character actions, which enables both the emergence of concrete storylines and the identification of instances later on.[7] While *Hennepin*'s architecture is fully implemented, I have not yet completed a second-order media experience that is driven by it; I am currently developing one in the form of *Sheldon County*, the subject of the next chapter. Thus, while in the chapters on *World*

[6]During the initial period of exploration into the integration of *Talk of the Town* and *Ensemble*, then undergraduate Joyce Scalettar was an important collaborator who sketched out valuable designs for potential emergent action structures. Even though I did not end up taking that technical approach, her work was important to the trajectory of the project.

[7]This method also enables the emergence of (improbably) interlocked storylines, which is the aesthetic hallmark of this simulation engine.

and *Talk of the Town* I could describe static versions of the simulation engines that were used to drive completed experiences, this is currently not possible with *Hennepin*. Due to the nature of feedback between the development of a simulation engine and an experience that it is meant to enable, *Hennepin* will almost certainly change in the future, but in this chapter I will describe it in its current state (at the time of this dissertation being filed).

Before proceeding, I should clarify my approach to evaluation. Because much is in flux, it is not possible to appraise *Hennepin* like I did my earlier simulation engines: through the lens of a completed second-order media experience. In lieu of such appraisal, in this chapter I will situate *Hennepin* against related technical work. For clarity, I discuss this related work in a distributed manner, throughout the chapter, as the pertinent concepts arise in my discussion. Finally: *Hennepin* is named for the county in which I grew up—Hennepin County, Minnesota—and this system is being reported for the first time here.

11.1 Modeling

Each *Hennepin* storyworld is modeled as an American county that is populated by several hundred characters (at any time), who purchase and subdivide land, start towns, establish businesses, construct buildings, produce artifacts, interact with one another, and contemplate on their lives. To develop this simulation engine, I started with the *Talk of the Town* codebase and modified certain elements according to the design goals that I outlined in the opening of this chapter, namely my desire to model fine-grained character actions. As such, much of the code from my earlier engine remains as a technical substrate, which means I have already explained some of *Hennepin*'s modeling. In this section, I will describe new features that are introduced with this engine and will point back to my earlier descriptions of components that have been directly inherited from *Talk of the Town* (and, in some cases, *World*).

11.1.1 Time

As in its predecessor *Talk of the Town*, time in a *Hennepin* storyworld proceeds by *day* and *night* timesteps, not all of which are actually simulated. While there is no explicit modeling of time within a timestep, character actions occur one at a time, and so there is an implied temporal ordering of simulated events.

11.1.2 Characters

Hennepin characters are generally modeled in the same way as *Talk of the Town* characters, though with some modifications that reflect my growing inclination toward a notion of character *believability* that borrows more from media tropes than from social science. I will tell the story of that evolution in Section 11.1.2, since it is particularly exemplified by the model of character personality that I have adopted in this system.

Name

Character *names* are generated using essentially the same procedure that is used in *Talk of the Town*, which I outlined in Section 9.1.2. As I noted there, a favorite detail of mine concerns the utilization of historical data that specifies the relative frequencies of the most common baby names through American history. Again, an additional forenames corpus is also drawn from in rare cases. This is the very corpus that drove *James Ryan Generator*—it is a thread knit through the entire history of my computational practice. *Hennepin* differs from its predecessor in terms of the naming procedure by incorporating new concerns into the probabilities that determine whether a child will be named for a family member, or given a hyphenated surname. Here, the new character personality traits and value systems, each of which I will describe shortly, are at play. For example, high scores for the trait `dutifulness` and for the value `tradition` will increase the likelihood of naming a child for a family member.

Appearance

The modeling of character *appearance* is likewise inherited outright from *Talk of the Town*, and so this aspect of *Hennepin* has already been explained at length in Section 9.1.2. As I discussed in that section, the curiously detailed modeling in this subsystem originated in the design of *Talk of the Town* (2015), a videogame in which character appearance was going to be central. While I do not envision appearance being particularly important in *Sheldon County*, I did not see the point of abandoning this aspect of the codebase, since the modeling is very lightweight and character appearance could conceivably be useful.

Personality

In developing *Hennepin*, I decided to overhaul the modeling of character *personality* that had been intact since *World* by adopting a 50-trait typology that was developed somewhat recently by Tarn Adams for *Dwarf Fortress* [17]. This design decision exemplifies a shift in inclination toward a notion of character believability that is more influenced by media tropes than by social science. In this section, I will tell the story of this personal evolution, which transpired gradually over the course of my tenure as a PhD student at UC Santa Cruz.[8]

As I explained in Sections 7.2.2 and 9.1.2, my earlier simulation engines uti-

[8] To jump to technical details about character personality in *Hennepin*, skip to the end of this subsection.

lized the five-factor ('Big Five') model of personality [810, 1337]. This utilization originated in a period of initial development work on *World* that preceded my time at UC Santa Cruz. After I arrived at the Expressive Intelligence Studio, I discovered that my coadvisor Michael Mateas is a critic of the Big Five as a model of character personality in procedural narrative. The crux of his critique is that the personality space that it yields is too low-dimensional to encompass the variety of characters we encounter in media. Here, I quote from Michael's own dissertation:

> Suppose we want to build James Bond using [a model of personality]. First, we would think about characters in the abstract. What general theory captures the notion of character? How might this general theory be parameterized (perhaps through infusions of "knowledge") to select specific characters? To inform the work, we might look at the dimensions of personality as described by various personality models in psychology (e.g. introvert–extrovert, thinking–feeling, intuitive–sensing, judging–perceiving). Once a generic architecture has been built, we could then define different characters by setting the right personality knobs. The problem is, how many of these personality knobs are needed to "tune in" a large number of characters? How many personality knobs need to be turned, and how many degrees of freedom does each knob need, in order to allow the creation of Bugs Bunny, Hamlet, The Terminator, Bambi? The differences between these characters seem to far outweigh any similarities. [764, p. 11]

Though this line of argument is compelling, especially when it is delivered in person, I initially pushed back against the idea of abandoning the Big Five in my own work. As I have articulated above, I viewed my work—especially on *World*—as being an exploration of authentic human experience. I was targeting the emotional experience of doing genealogical research, reading a generational narrative, hearing someone's personal testimony, living out a life, having deathbed regrets. With these aims, I had no use for Bugs Bunny or Bambi or The Terminator. Since the Big Five is a developed account of human personality, I thought it would make a good model for the kinds of characters that I sought to develop. Moreover, since there is extensive research on how the Big Five model correlates to a huge variety of human phenomena, it made modeling the causal factors for such phenomena (as rooted in personality) an easy matter of operationalization. For instance, as I noted in Section 7.2.2, I could model the formation of friendship (as factored by personality) in a matter of minutes by looking up a paper and rendering its findings in code.

Over time, however, I came to terms with a simple fact: I was making media. Of course, I knew this all along, but I had not developed artistic aims for my practice accordingly. My simulated worlds are tiny and abstract—everything, even space and time, works differently in these worlds than in the real world—and though the little computer people that live in them may have had a capacity to evoke a kind of authentic human experience, this was more likely an effect of *believability* than of realism. Here is another quote from Michael's thesis:

> Believable agents are the union of AI-based autonomous agents and the personality-rich, emotive characters that appear in the dramatic arts (e.g. theater, cinema). "Believability" is a term used by character artists to describe a property of compelling characters, namely, that of engaging in internally consistent, lifelike and readable behavior in such a manner as to support an audience in suspending disbelief and entering the internal world of the character. This is not the same as realism. Characters are *not* simulations of people, exhibiting behavior indistinguishable from humans under some controlled laboratory condition. Rather, characters are *artistic abstractions* of people, whose behavior, motivations, and internal life have been simplified and exaggerated in just such a way as to engage the audience in the artist's vision. [764, p. 8]

For me, this distinction connects back to my discussion of simulation in Section 4.1.5. One way of characterizing my earlier approach to my simulation practice is that I viewed my style as being representational, when in fact it would be more aptly described as *impressionistic*. My characters are artistic abstractions of people.

At the same time I was reimagining my simulation practice as being primarily an effort in media creation—and, moreover, having impressionistic qualities rather than more concrete representational ones—with Ben Samuel and Adam Summerville, I was performing *Bad News* somewhat frequently. While this project provided a lot of authorial knowledge on how to do story sifting, as I discussed in the opening of this chapter, it also lent insight in terms of character modeling. As an actor tasked with playing any character that might emerge in a *Talk of the Town* storyworld, Ben had to develop methods for rendering Big Five personality models in fully embodied portrayals. While his character performances are marvelous, and while he has been gracious in his appraisal of the personality modeling of *Talk of the Town*, I felt that the simulation engine was not giving him a lot of material with which to work.

Because it includes only five traits and models the distribution of values for those traits according to reports of such distributions among real people, *Talk of the Town* tends to produce characters with neutral personalities. As Ben elucidates in his own dissertation [1084, pp. 370–371], his techniques for modulating posture, voice, and other performance dimensions hinge in large part on a character's strongest traits. As such, characters with no strong traits offer little direction for how they should be played, and characters having a single extreme trait are defined rather one-dimensionally. Moreover, when it came to how personality might drive behavior in more specific domains—at work, in romantic affairs, among family—Ben was faced with the challenge of reasoning about the entirety of human phenomena in terms of only five abstract factors. As a gifted actor, he could fill in the gaps improvisationally to maintain believability, but his task was harder than it had to be, and I certainly could not trust fully computational characters to maneuver so adeptly. Because the impetus for this new simulation engine was to ramp up the simulation granularity—to model fine-grained character actions—I could no longer rely on an abstract personality model being good enough to support abstract character behavior.

Thus, in my personal journey with personality modeling, I had come to internalize the essence of Michael Mateas's critique of the Big Five model: it has too few dimensions to account, computationally, for the variety of character phenomena we find in media. It should be noted, however, that this is not the extent of his critique. In fact, Michael is questioning the plausibility of using character personality models in a general sense. As he explains in his own dissertation (and elsewhere [758]), Michael operates in the intellectual and technical tradition of the Oz Project [83, 84], which espoused a direct approach to character authoring that precludes the latency inherent in a personality model. At its most concrete, this approach entails the direct authoring of a pattern of character *behaviors*, so that the character's personality (and moreover its every facet) is expressed through concrete behavior in the simulated storyworld. This is how character authoring worked in the Oz Project, as enabled by the Hap agent language developed by that group, as Michael explains:

> Hap provides mechanisms that support writing behaviors for characters. A behavior is a chunk of activity; such behaviors can be high-level (a behavior for "having fun"), or low-level (a behavior for moving the character's body when they open a door). If you wanted to build James Bond in Hap, you would identify high-level goals (motivations) that make James Bond who he is. Then you would think of the multiple ways (behaviors) that James Bond might use to accomplish these high level goals. These multiple behaviors probably themselves have subgoals. Any given behavior is only appropriate under certain conditions (what's recently happened, how Bond is feeling, what's happening right now in the world, etc.); these conditions are captured within each behavior. At every level of description, James Bondness can be

> infused into the character. From how Bond thinks, to how Bond walks, the artist has the control to create the character consistent with their vision. [758, pp. 11]

This mode of character authoring also characterizes the ABL agent language that was used by Michael and Andrew Stern in the creation of *Façade* and the later *IMMERSE* project, which was a larger collaboration that included several other cocreators at UC Santa Cruz and BBN Technologies [335, 1140, 1139, 1138].[9]

Is this how personality works in *Hennepin*, then? Do I author character behaviors directly, rather than latently through the mechanism of a personality model? No, I do not. As a student of Michael Mateas, who was himself a student of its figurehead Joe Bates, I also operate in the intellectual and technical tradition of the Oz Project—but I believe that my practice would actually be undermined were I to attempt to author character behaviors in this direct manner. There are two reasons for this, each of which connects back to the *aesthetics of emergent narrative* that I articulated in Chapter 3.

First, in pursuit of the *aesthetics of the unathored*, the *aesthetics of the ephemeral*, and the *aesthetics of the actual*, I am reluctant to do anything but procedurally generate characters. If I were to create *Hennepin* characters through a process of directly authoring the sets of behaviors that constitute those characters—how Grace and Trip were created for *Façade*, for instance, modulo considerations of joint behaviors [779]—then each *Hennepin* storyworld would have the same char-

[9] ABL is a descendant of Hap whose fundamental design change was to support the authoring for *joint behaviors*: compositions of interlocking individual agent behaviors, structured around the conceptual pattern of the *dramatic beat* [774, 773, 777]. In Section 11.2.7, I discuss how the essence of joint behaviors has made its way into *Hennepin* through an intellectual and technical tradition that connects back to ABL via the mediating force of another AI system, *Comme il Faut* [804]. An additional note: interestingly, the Oz approach to character authoring is also, more broadly, an approach to character representation itself. An Oz character is not just driven by a collection of authored behaviors, but moreover it *is* that collection of authored behaviors (along with some instantial assets by which the character may be represented visually, in the case of a graphical world).

acters. This would undermine the aesthetics of the unauthored, since lovingly handcrafted characters, though potentially more believable and more compelling, are not unauthored.[10] Relatedly, such an approach would compromise the aesthetics of the ephemeral, which characterizes worlds that have never existed before and will never exist again. If you turn off a simulated world and turn on the next one only to find the same characters living in both, then there is no meaningful ephemerality (in the sense of the notion that is under discussion here).[11] Finally, recall that the aesthetics of the actual stands in opposition to the trappings of fiction, particularly in terms of the notion of a simulacrum. As I note in the next chapter, a prelude to each *Sheldon County* pilot episode contains this segment:

> Each episode in this series will be procedurally generated according to the particular characters and stories that have emerged in your county. These characters that you will come to know are not simulacra. They are not impoverished representations of real persons. They do not exist more perfectly in the mind of a human author.

As I have noted above, I contend that the characters of fiction are in this sense a kind of simulacrum: they exist more perfectly in the minds of their authors. But when characters are procedurally generated, they only exist in the worlds that bear them, and they do not stand for anything else—they are *actual*.

The second reason for my deviation from the Oz Project tradition of direct character authoring is more practical: my simulated storyworlds are relatively massive in terms of the number of characters, so even if I were willing to abandon the procedural generation of characters, it would be infeasible for me to manually

[10]Of course, there is a spectrum here, and Hap and ABL characters are certainly not on the far end of it where, for example, the static characters of novels might reside. Though such characters rely on authored behavior patterns to act, they are still dynamic entities that behave emergently (in ways that could even surprise their authors). In this sense, such characters are to some degree procedurally generated. But they persist across runtime instances, and this is the critical distinction under discussion here.

[11]Again, we can introduce nuance here: all interactive media—no, *all experience*—is ephemeral, thus these aesthetics are implicit in all human phenomena. But this quickly devolves into pedantry.

author so many examples. As I have articulated in detail above, I create worlds with many characters to support the *aesthetics of the vast*, and moreover as a way to overcome some of the pitfalls of emergent narrative that were identified in Chapter 4. Thus, for my practice, the avoidance of direct character authoring is both an artistic choice and a practical one.

Of course, I have already divulged my disenchantment with the Big Five personality model, so where is the middle ground between that and something like the Oz approach to character authoring?[12] It would have to be one in which characters are represented according to abstract patterns that may be composed at runtime, which themselves may be used to generate character behaviors. This was the case in my utilization of the Big Five personality model in *World* and *Talk of the Town*, but I did not find it to be rich enough. As such, my solution has been to simply use a richer personality model, by which I mean one with more traits, and whose traits are more evocative.

In *Hennepin*, I utilize the personality model that is being used in the current version of the *Dwarf Fortress* [17].[13] This typology was developed by its cocreators Tarn and Zach Adams over the course of multiple iterations that appeared in earlier versions of the game, each of which is documented on a fan-maintained Wiki [278, 286, 287, 280]. Like me, the Adams brothers also began with the Big Five traits, but instead they used the expanded *NEO PI-R model*, which unpacks the five factors into a set of thirty total traits that are more specific [216, 278]. As Tarn told me recently in conversation, he and his brother found that the personality models found in psychology are too generous to support fundamental kinds of

[12] Interestingly, Michael positions the Oz approach itself at the center of a continuum between top-down approaches driven by, for example, personality models, and the bottom-up approaches of artificial life that might rely on evolutionary computation or neural networks [758, p. 10].

[13] At the time of writing, this is `v0.44.09`. I will note also that I adapt this model with its creators' permission (personal communication, April 17, 2017).

narrative intrigue.[14] For example, the Adamses sought a model whose possibility space would encompass conventionally evil characters, but psychologists tend to refrain from crafting personality tests that may conclude that a respondent is evil. To rectify this gap, the *Dwarf Fortress* creators turned, intriguingly, to the writings of Thomas Aquinas [47] (and others) to expand their model. Here is how Tarn explained this process in a recent personal communication:

> In the earliest versions of Dwarf Fortress, dwarves were distinguished by a number of boolean traits and indexed preferences. Do they like the outdoors? Are they scared of spiders? What are their favorite foods? When we first felt the need to distinguish broader personality facets for the dwarves numerically, it was natural to hunt around for an existing system that looked like it had enough resolution for our purposes. The version of the Big Five [...] with 30 facets seemed promising, but it soon became clear that it was designed for real people rather than characters in a story. Characters in stories can be complex, but they often have clearer virtues and flaws. It's also okay and sometimes desirable to judge them on reasonably stark terms, and highlighting them in these terms is important if the player is to understand what's going on, especially in a game without a pre-authored narrative [...] This led us to St. Thomas and Summa Theologica, in part, as well as a number of dramatic character archetype classifications and so forth (these were all internet searches, and sourcing was not important, so I don't remember the specifics of the web dive.) The Big Five system still had a lot to offer, so we formed a very non-academic, non-theological amalgam of the various systems we'd encountered, trying to keep the facets as independent as possible. This succeeded in making the results more satisfying.[15]

The fifty traits in *Dwarf Fortress*'s personality model [280] are as follows:

> `abstract inclined`, `activity level`, `altruism`, `ambition`, `anger propensity`, `anxiety propensity`, `art inclined`, `assertiveness`, `bashful`, `bravery`, `cheer propensity`, `close minded`, `confidence`, `cruelty`, `curious`, `depression propensity`, `discord`, `disdain advice`, `dutifulness`, `emotionally obsessive`, `envy propensity`, `excitement seeking`, `friendliness`, `gratitude`, `greed`, `gregariousness`, `hate`

[14] Personal communication, March 23, 2018.
[15] Personality communication, May 3, 2018.

`propensity`, `hopeful`, `humor`, `imagination`, `immoderation`, `immodesty`, `love propensity`, `lust propensity`, `orderliness`, `perfectionist`, `perseverance`, `politeness`, `pride`, `privacy`, `single minded`, `stress vulnerability`, `swayed by emotions`, `thoughtlessness`, `tolerant`, `trust`, `vanity`, `vengeful`, `violent`, and `wastefulness`.

While Tarn fits each trait to a scale spanning between `0` and `100`, I use an adapted scale that spans from `-50` to `50`. Following Adams, I informally quantize this range in the following way (for authorial scaffolding):

- `50` to `41`: `extremely high`
- `40` to `26`: `very high`
- `25` to `11`: `high`
- `10` to `-10`: `neutral`
- `-11` to `-25`: `low`
- `-26` to `-40`: `very low`
- `-41` to `-50`: `extremely low`

One benefit of using the *Dwarf Fortress* personality model is that its creators have written glosses for each of the quantized value ranges of each trait, aside from the neutral range. These glosses specify how a character's behavior may be driven according to her degree of activation for a trait, and conveniently they are available on a fan-maintained Wiki [280]. For example, here are the glosses for the trait `friendliness` (fit to my adapted value range):

- `50` to `41`: *is quite a bold flatterer, extremely friendly but just a little insufferable*
- `40` to `26`: *is very friendly and always tries to say nice things to others*
- `25` to `11`: *is a friendly individual*
- `-11` to `-25`: *is somewhat quarrelsome*
- `-26` to `-40`: *is unfriendly and disagreeable*
- `-41` to `-50`: *is a dyed-in-the-wool quarreler, never missing a chance to lash out in verbal hostility*

As I note below in Section 11.2.7, the Adams brothers' trait glosses (and the traits themselves) provide authorial scaffolding that in my project has worked to facilitate the process of defining character actions. For instance, one authoring pattern has worked according to the following process: read a gloss for an extreme trait quantization (either positive or negative), think of a discrete character action that would express that kind of extremeness in my simulation domain, and author an action accordingly (whose preconditions depend on a character's value for that trait falling in the extreme range). Thus, for example, a *Hennepin* character possessing extreme `vanity` may perform the action `admire-reflection-in-mirror`. Even if I authored just one action for each trait polar extreme, that would still result in one hundred character actions. Of course, actions may also be authored for other trait value ranges, though they may be less salient than the ones authored for extreme ranges. Beyond facilitating the process of action authoring (by authorial scaffolding), this pattern leads to potent actions that satisfy one of Richard Evans's desiderata for procedural character personality:

> The second major requirement is that each personality trait has a distinct and obvious effect on autonomous behaviour. If you create a character that is a foul-mouthed extrovert, and leave him to his own devices, he should go out meeting new people and swearing at them. Further, the way he autonomously manifests his personality should be transparent to the player: the player shouldn't have to take careful notes of every action the character does for many days, compile them in a spreadsheet, in order to notice that this character is 11.3% ruder. Manifestation of personality has to be obvious in individual behaviour without recourse to statistical patterns. [325, p. 35][16]

Additionally, multiple actions could express the same extreme trait in different social contexts: thus, a character possessing acute `cruelty` might `ridicule-a-family-member` at home, `scold-an-inferior` at work, or `berate-an-employee`

[16]In developing *Versu*, Evans and collaborator Emily Short operated under a core design goal of authoring actions that vividly express personality [326, p. 118]. This project was preceded by *The Sims 3* [1207], in which Evans also pursued this aim [324].

at a store while on an errand. Moreover, there is a massive combinatorial space suggested by the potential interaction of traits, and accordingly I have also authored actions by considering how two extreme traits could interact in ways that character behavior might express. Though I have not created one, I can imagine a powerful authoring assistant that would suggest which actions to author by simply combining the glosses for two different trait ranges—this would be similar to the author-assistance visualizations that aided Jacob Garbe and Aaron Reed in the crafting of procedural content for *The Ice-Bound Concordance* [377, 985].[17]

It is worth comparing *Hennepin*'s modeling of character personality to that of *The Sims 3* [1207], a videogame on which Richard Evans served as AI lead. In the game, each character ("Sim") is attributed up to five personality traits, from a total collection of eighty; example traits include `inappropriate`, `grumpy`, `over-emotional`, `slob`, and `kleptomaniac` [324]. As in *Hennepin*, traits may gate certain actions, according to authored condition logic, and additionally they can work as "adverbial modifiers" [324, n.p.] that alter how a Sim performs a common action. For example, a clumsy character is likely to trip while performing `walk`, while a workaholic may pull out her phone in carrying out `wait`. In *Hennepin*, I have likewise authored personality-driven variants on common actions—for instance, a character with low `ambition` might perform `slack-off-at-work`, while someone with the opposite trait value could `work-hard-at-work`.

While characters in *Hennepin* are assigned values for all fifty traits, a given character is only likely to possess a handful of extreme trait values. Since neutral trait values are generally ignored (aside from being inherited by offspring), a *Hennepin* personality profile actually functions quite similarly to the eighty-pick-five profiles that are attributed to characters in *The Sims 3*—that is, since

[17] This visualization framework was developed in collaboration with Melanie Dickinson, who was then an undergraduate at UC Santa Cruz, but it was intended specifically to aid the authoring work on the project that was carried out exclusively by Garbe and Reed.

only extreme traits are important, each *Hennepin* personality profile effectively comprises on the order of five or so traits.

Finally, while Sims choose what to do in accordance with explicitly modeled *motives*, in *Hennepin* such motives are implied in the condition logic for character actions. For instance, while in *The Sims 3* a mean-spirited Sim's motive to torment others will figure in the system's utility-based procedure for action selection [324], in *Hennepin* a cruel character will likewise be more likely to torment others, but simply by virtue of the condition logic on such actions necessitating an extreme trait value for `cruel`.[18]

The generation of a personality profile for a given character works according to the same principles used in *World* and *Talk of the Town*. A character's personality is built up one trait at a time, and whether a given trait will be inherited from a parent depends on heritability probabilities specified for each trait. (Of course, characters without parents cannot inherit any trait values.) If it is determined that a trait will be inherited by a child, the child takes after one of the parents, selected randomly, but with some variation. This works by utilizing a pseudorandom number generator whose outputs fit a normal distribution, given a mean and standard deviation. Specifically, the inherited value for that trait—the trait value of the parent from whom the trait will be inherited—is used as the mean and a trait-specific *heritability noise factor* (specified in a configuration file) is used as the standard deviation. If it is determined that a trait will not be inherited

[18]This distinction is rooted primarily in the disparity in the number of characters modeled in the systems' respective storyworlds: in *Hennepin*, worlds are populated by upward of a thousand characters over the course of simulated centuries, while a given *Sims 3* town is populated by a few dozen Sims for a much shorter duration of story time. As I discuss below, due to the number of characters—and especially the detail with which each is modeled and simulated—it is not feasible for me to score all candidate actions during action selection. As such, there would not be much sense in building out a full-fledged motives system, since it would likely not end up figuring much into action selection. That being said, I do model character aspirations, though these currently do not figure into action selection.

by a character—for instance, because she has no parents—then the same normal-distribution pseudorandom number generator is used, but instead according to a *trait population mean* and a *trait population standard deviation.* All trait values are automatically constrained to fall between `-50` and `50`.

In earlier work, I strove for realism in my modeling of character personality, but over time I began to wish for a more evocative framework that could better yield narrative intrigue. While this was partly a function of encountering inadequacies in the storyworlds produced by my simulation engines, it was also due in large part to the influence of my coadvisor Michael Mateas, and the larger intellectual and technical tradition to which we both belong. In the Oz Project, Joe Bates and his collaborators aimed to build autonomous characters that worked not like human beings, but like the characters we encounter in media. As such, its core directive was to seek inspiration primarily from the *character arts* [1247], as opposed to the social sciences or other fields of human study [83, 87, 84, 991].[19] While my technical method for character modeling does not resemble those which underpin the Oz Project and later work by Michael Mateas and collaborators, in *Hennepin* my approach is now couched in the same intellectual tradition. I still seek to generate stories with heart, about authentic human experience, but for now I

[19]Though seminal in this regard, the Oz Project was not the only group that found inspiration in the character arts. Likeminded contemporaneous efforts include the *Petz* and *Babyz* videogame series produced by PF Magic (which included Andrew Stern) [352, 1201, 1199], Millennium Entertainment's *Creatures* series [414], and several more [549, 120, 758, 466, 1198]. In the late 1990s, these practitioners converged yearly on the nascent International Conference on Autonomous Agents, which in 2002 merged with two other conferences to form the International Conference on Autonomous Agents and Multi-Agent Systems, causing the expressive tradition rooted in the character arts to lose its academic home. For a great overview of this period, see the contemporaneous reviews of Clark Elliot [307] and Michael Mateas [758] or the annotated bibliography of Barbara Hayes-Roth and Patrick Doyle [464]. Later, at the 2002 AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, there was an interesting debate on the relative merits of realism and believability [409]. Finally, as Rob Zubek notes [1382, p. 10], the latter notion relates to Masahiro Mori's notion of the *uncanny valley* [855]—the famous English phrase being due to Jasia Reichardt's translation [990]—since caricature is a good method for avoiding the valley.

plan to do this with characters who work more like the ones we find in media. It is possible that later on I will view this move as swinging the pendulum too far toward the other side, but for now I am content and excited.[20]

Value System

Hennepin characters form and dynamically reconstruct *value systems* that capture opinions about what is important in life. Like the personality model described in the last section, my value typology was also designed by Tarn and Zach Adams for *Dwarf Fortress* (and is likewise used with his permission). Unlike his personality model, this system (called the 'beliefs' system [280]) is just starting to be incorporated into the game in a significant way.[21] As Adams told me recently in a personal communication, the development of this model was more haphazard:

> we found we needed to separate the more elemental personality from later learned intellectual values; this process is newer and has been less rigorous. We just wrote down a list of several values without attempting to be exhaustive; we assumed any philosophical system we added later would eliminate our simple work here, and the seeds of this change have been planted with the game's philosophical knowledge 'forest' (which is also simple, but in some ways orthogonal to the existing value set; what we'll end up with is still a mystery to me).[22]

Though its development may have been relatively hasty, I find the Adams brothers'

[20]I should note that these two approaches—operationalizing ideas from the social sciences about how people work and modeling character tropes found in media—are not mutually incompatible. For example, in his collaborative work on *CiF* and *Prom Week*, Josh McCoy (who has the same academic lineage as me) has done a nice job of managing an admixture of inspiration from the social sciences (Goffman's *dramaturgical analysis* [399]) and media tropes [807]. More immediately, the very personality model I just described pulls its traits in this way from distinct cultural purviews, and other aspects of *Hennepin*'s architecture and authored procedural content are still influenced by the social sciences (as the next section will show). My pendulum swing thus brings me closer to the other side, but not all the way. In a recent conversation, I was surprised to hear from Michael Mateas that behind the scenes in the Oz Project, there was actually more emphasis on realistic agent modeling than the group's papers suggest. To me, it seems that a mixture of these concerns is not only natural, but perhaps also inevitable when it comes to the task of actually building autonomous characters.

[21]Personal communication, March 23, 2018.

[22]Personality communication, May 3, 2018.

model to be quite inclusive, and also dramatically potent—as I noted in the last section, such potential provides critical authorial scaffolding for action authoring.

What follows are the thirty-four elements in this model, each of which I call a *value* (again, in *Dwarf Fortress* these are termed 'beliefs'):

> `artwork`, `commerce`, `competition`, `cooperation`, `craftsmanship`, `cunning`, `decorum`, `diversity`, `eloquence`, `fairness`, `family`, `friendship`, `hard work`, `harmony`, `independence`, `introspection`, `knowledge`, `law`, `leisure time`, `loyalty`, `martial prowess`, `merriment`, `nature`, `peace`, `perseverance`, `power`, `romance`, `sacrifice`, `self control`, `skill`, `stoicism`, `tradition`, `tranquility`, `truth`

As in the *Hennepin* personality model, I use a numeric scale spanning from `-50` to `50`, in this case to represent a particular character's opinion about the value. In this way, a score of `-50` for `decorum`, for example, represents a case of a character who does not value decorum in the slightest. Alternatively, on the other end of the spectrum, a character with a score of `50` would value decorum to the utmost. Here, I use same quantization scheme as before:

- `50` to `41`: `extremely high`
- `40` to `26`: `very high`
- `25` to `11`: `high`
- `10` to `-10`: `neutral`
- `-11` to `-25`: `low`
- `-26` to `-40`: `very low`
- `-41` to `-50`: `extremely low`

Just as they did for their personality model, the Adams brothers have written glosses for each value that specify how a character's behavior may be driven according to her degree of fervor for that value. Conveniently, these glosses are available on the same fan-maintained Wiki [280]. As an example, here are the glosses for the value `decorum`:

- `50` to `41`: *views decorum as a high ideal and is deeply offended by those that fail to maintain it*
- `40` to `26`: *greatly respects those that observe decorum and maintain their dignity*
- `25` to `11`: *values decorum, dignity and proper behavior*
- `10` to `-10`: *doesn't care very much about decorum*
- `-11` to `-25`: *finds maintaining decorum a silly, fumbling waste of time*
- `-26` to `-40`: *sees those that attempt to maintain dignified and proper behavior as vain and offensive*
- `-41` to `-50`: *is affronted by the whole notion of maintaining decorum and finds so-called dignified people disgusting*

As I will explain below, one of my authoring patterns for specifying action definitions involves reading the extreme variants of these glosses and imagining what kind of action a *Hennepin* character with such an opinion would take. Beyond providing authorial scaffolding that facilitates the process of creating new action definitions, this pattern is powerful because it leads to character actions that are not only believable, but moreover vividly manifest a character's opinions on life. Again, this accords with Richard Evans's call for character behaviors that express who those characters are [325]. Critically, character values also drive character *aspirations*, a notion that I will explain in the next section. For example, a character who believes strongly in both `law` and `power` may form the aspiration of wanting to become a police officer. Lastly, I would like to note related work, by Sasha Azad and Chris Martens, that has resulted in a system in which characters cultivate and propagate opinions [56].

Aspirations

In *Hennepin*, characters may form *aspirations* about what they would like to accomplish in life. As I will explain below in Section 11.2.8, aspirations are defined to include conditional triggers that, if fired for a given character, cause her to form the aspiration. Generally, these conditions depend on the character's personality and also her value system. In turn, another trigger captures what must be true about the storyworld for the aspiration to be fulfilled. This component of *Hennepin* is inspired by the character goals of *Universe* [663, 665] and the *wants and fears* system of *The Sims 2* [793, 146, 1162].

Knowledge

Whereas the rich modeling of character *knowledge* is the hallmark of *Talk of the Town*, character actions are the central concern in *Hennepin*. As such, I elected to start over from scratch to develop a more lightweight approach to simulating character knowledge in this engine, since I now find *Talk of the Town*'s modeling of such phenomena to be both somewhat bloated (from a computational standpoint) and somewhat burdensome (from an authorial standpoint).

In *Hennepin*, characters form knowledge about what has happened in the past, as opposed to beliefs about other people and places. This is a course correction meant to rectify what I now perceive to be a blunder in *Talk of the Town*'s knowledge modeling: the subject of character beliefs are too boring to be narratively potent. For example, while the image of an entire town collectively harboring false beliefs about a reclusive character's hair color may be amusing, it would be difficult to construct an engaging story recounting these circumstances.

As I will explain in more detail in Section 11.2.9, a *Hennepin* character's collected knowledge of the world is structured as a collection of *knols*, each of which represents knowledge of some action that occurred in the past. A 'knol' is a term meaning 'unit of knowledge' that was coined by workers at Google in 2007 [3]. Each *Hennepin* knol maintains a list of its *sources*, which are all the actions that imparted knowledge about the subject of the knol (more on this later). The system also keeps track of knols that a character has forgotten, but in this system characters do not misremember (as a simplifying measure). Additionally, knols are indexed according to tags that capture characteristics of a remembered action. For example, a tag `embarrassing` may be attached to a knol about something embarrassing that happened to a character, which allows the system to retrieve it accordingly if some other action calls for such a knol (for example, an action `cringe-over-embarrassing-memory`).[23] Finally, characters build up simple probabilistic models about the locations of other characters that rely on known past locations. This allows one to track someone else down in the case of an emergency, for example, as I will explain in Section 11.2.9.

Miscellaneous Attributes

As in *Talk of the Town*, characters in *Hennepin* are also modeled according to several additional miscellaneous attributes, including *sex*, *sexuality*, *fertility*, and *memory*, the latter of which is again modeled as a floating-point value that decreases over time, this case making forgetting more likely. Likewise, the system also tracks character-specific data pertaining to a number of concerns including social relationships, romantic history, work history, residential history, and more.

11.1.3 Other Entities

In addition to characters, *Hennepin* models a county (the storyworld), along with its towns, plats, parcels, lots, roadways, buildings, homes, businesses, artifacts, and statutes. I will briefly describe each of these in this section.

[23] This is inspired by the tagging of past actions recorded in the *social facts database* of *Comme il Faut* [804, p. 101], the AI engine that drives *Prom Week* [799, 803].

Counties

A *county* in *Hennepin* is modeled as a square grid platted according to the procedures of the United States Public Land Survey System [1328]. Each county comprises four *survey townships*, two on top and two on bottom. A survey township is itself a grid that spans six miles by six miles and is composed of 36 square-mile portions called *sections*.[24] A section is a 640-acre portion of land that is composed of four *quarter sections*, each of which has an extent of 160 acres. A quarter section is the classical slice of American land, since it was the portion that was given away by the United States General Land Office, according to the Homestead Act of 1862, to white settlers as an enticement to bring them west to till the now vacated lands that had been formerly possessed by various Native American groups [951].[25] When a quarter section was claimed as a homestead, the claimant would typically build a farmhouse in the middle of the plot, which is the origin of the Americanism "back forty", denoting a remote part of the farm located behind one's house [691].

As I will discuss below, in *Hennepin* the Homestead Act is loosely modeled by generated characters coming to the county to claim quarter sections and establish farmsteads. Initially, however, the county is represented as a grid of townships, each being represented as a grid of sections, and subdivision only occurs as character actions demand. While I am now exploring the integration into this project of an earlier terrain generator that I originally developed for an abandoned extension to *World*, as mentioned above, the counties currently do not model terrain.

[24]A survey township is not to be confused with a township, which is a type of municipality in many states. Often, townships are coextensive with survey townships, but the latter were typically platted prior to any actual municipality being established there.

[25]Beginning in 1866, land grants became available for certain people of color as well, though to an extent this was nominal since institutional policies made it difficult for such citizens to actually make use of the program [901]. Such institutionalized racism was a factor in keeping a majority of black Americans in the southeast portion of the country until the so-called *Great Migration* of the twentieth century [439].

Parcels, Lots, Roadways

Each portion of land in a county is called a *parcel*, which is a generalized notion that extends the earlier modeling of tracts and lots in *Talk of the Town*. Parcels vary in size as land is subdivided according to character actions. For instance, a character might offer to purchase half of another character's parcel, in which case that portion of land will be subdivided into two halves. Such breaking down may occur repeatedly over time, as in the real world, to create dynamics of settlement density where more populated areas are bounded by rural areas constituted in large parcels (likely used for farms).

The smallest unit of land modeled in *Hennepin* is the *lot*, which is a *square chain*, meaning it spans 66 feet on each side. Parcels are composed of one or more lots, and a lot is the unit of land on which a building may be constructed. At any point during the simulation, a character will be located either in a building on a lot, or on a lot itself. Thus, if a character is out in the middle of a secluded parcel, technically she will be positioned on one of its lots. In this way, lots work like the *tiles* that compose a *Dwarf Fortress* world [281], for instance, though *Hennepin*'s lots are much larger in size. Each county's lot grid comprises 921,600 lots in total.

Whereas in *Talk of the Town* each *roadway* system was instantiated chunk by chunk on a predetermined grid, in *Hennepin* these emerge naturalistically as interconnections between buildings are needed. To request roadway access, a character performs an action that results in a request being submitted either in writing or in person to officers in the county courthouse. The roadway path is then determined by a simple $A*$ procedure [1036, pp. 96–101] that searches in the space of possible paths on the grid of parcel perimeters, according to some simple heuristics (namely to prefer straighter paths). An actual roadway is then constructed piece by piece by county laborers, who perform actions like `pave-section-of-road`.

As I note below in Section 11.3, while such actions may appear to be benign, they could actually tie in with other systems to yield significant narrative intrigue. For instance, characters may bury artifacts on lots, and an action like `pave-section-of-road` could have the effect of unearthing such objects. Generally, a major goal in developing this system is to hook infrastructural modeling into other systems so that the former may actually produce narrative intrigue.

Towns

While the initial characters who move into a county will live in an unincorporated area, over time characters may establish *towns*, which are physically represented as collections of parcels. The formation of a town will often coincide with the platting of a *townsite*, which is a subdivision of a section (square mile) into an aesthetically pleasing arrangement of small *town lots*, planned roadways, and a town square. When a character purchases a section and subdivides it according to a townsite plat, she forms a land company that sells those lots to other characters who would like to live there. The modeling of municipal evolution in *Hennepin* represents a return to the *settlement* phenomena of *World*.

I decided to model the larger area of a county—rather than a town only, as in *Talk of the Town*—in part because I wanted more space so that emergent character activity could become reified in the storyworld topography itself. This provides for narrative intrigue associated with a number of media tropes, such as the *town rivalry* and the *utopian settlement*. As for the latter, characters may collaborate to establish towns that are rooted in shared principals—for instance, a group of characters who all believe in `law` and all despise `merriment` may establish a town based on those shared values (more on this example below).

Buildings

In *Hennepin*, a *building* may be constructed on top of a lot as a result of character actions pertaining to that construction. Unlike in *Talk of the Town*, a building and a business that is headquartered there are decoupled as separate modeled entities. In every other way, a building is modeled following my previous simulation engine.

Homes, Businesses, Vocations

The *homes* and *businesses* of *Hennepin* are modeled just as they in *Talk of the Town*, except that the latter are now decoupled from the buildings in which they are situated, as I just noted. Additionally, whereas in the earlier system a business was modeled as either requiring a tract or a lot to operate, businesses in *Hennepin* are defined according to more specific size constraints that they place on the land on which they will operate. For instance, a farm requires considerable acreage, whereas a drug store may be established on a single lot.

Artifacts

Characters may perform actions that cause physical *artifacts* to exist in the storyworld. This system is an offshoot of the modeling of character actions, so I will hold off on explaining how it works until later, in Section 11.2.11.

Statutes

One system that is meant to support interesting feedback loops (between storyworld and emergent events) concerns character-defined *statutes* that may change how the storyworld works in a certain jurisdiction. Statutes are proposed and voted on by characters at county or town hall meetings, and they generally pertain to the prohibition of certain kinds of character activity. For instance, in the example town that I introduced above—founded on the common principals of law and no merriment—the characters might decide to ban alcohol there. As a result, drinking actions would become illegal and the police officers in the town would have a new social affordance of performing arrest actions on characters who are known to have broken that law. I will revisit this example in Section 11.3.

11.2 Simulation

Now that I have explained the basics of *Hennepin*'s entity modeling, in this section I will outline its simulation procedures. At a high level, the workflow here is the same as that of its predecessors *World* and *Talk of the Town*: create an empty world, have some initial characters move into it, and then proceed from there, timestep by timestep, simulating storyworld phenomena by executing a recurring simulation loop. By this configuration, there are two distinct modes of simulation, one pertaining to the initial setup of a storyworld and the other to its evolution over time. I will describe both of these in turn.

11.2.1 Setting Up a County

Each *Hennepin* storyworld begins with the same essential scenario, which resembles that of *Talk of the Town*: it is the summer of 1839 and there is an empty American county that has already been platted by the United States land office. Rather than a set of initial characters automatically moving into the storyworld right at the onset, in *Hennepin* time proceeds with it being empty, until a first character decides to move there according to one of several probabilistic rules that pertain to: drifters coming through the county, homesteaders claiming land there, and characters arriving to start other kinds of businesses. As with my earlier simulation engines, adult characters who are generated outside the storyworld undergo a retcon procedure that abstractly simulates a backstory.

A More Nuanced Evolution

Whereas in *Talk of the Town* the opening of a new employee position could cause the spontaneous arrival of a new character to take that job, in *Hennepin* jobs are filled more naturalistically. Specifically, when a job opens, the employer will make the opening known by word of mouth or by artifacts, such as job postings, that encode information about there being an open position. In this way, an employer may post about an opening at county gathering spots, and drifters passing through the county may decide to apply for a position and stay in the county. As such, the evolution of a *Hennepin* county occurs by characters taking fine-grained actions in the storyworld, with new characters coming into (or through) the county only as probabilistic rules trigger such arrivals.

Finally, as in *Talk of the Town*, the storyworld may be seeded, which means a county and its exact history will be reproduced when a seed is reused. In the case of *Sheldon County*, this could potentially support the practice of *seed sharing*, where listeners share seeds that they have found to produce interesting podcasts. This practice is popular among the *Dwarf Fortress* player community [285].

11.2.2 Simulation Loop

Once a county has been established, its history is simulated until a specified termination date, in the style of *Talk of the Town*; in the case of *Sheldon County*, I have not yet specified an end date.[26] As with *World* and *Talk of the Town*,

[26]Unlike *Bad News*, which takes place at the time of the termination date—in that case, the summer of 1979—*Sheldon County* has no diegetic frame story, which means it does not take

time progresses in *Hennepin* according to the execution of a simulation loop that recurs each timestep. As in *Talk of the Town*, not every timestep is actually simulated. On a timestep that is simulated, an *action manager* carries out its general procedure for *action selection*, whereby everyone in the county may take action. In other cases, if characters urgently need to take a *queued action*—such as giving birth on a preordained due date—then only those characters take action on the timestep. Figure 11.1 shows an example of the kind of raw console output that is displayed on screen as this procedure is underway; as such, it gives an example of the kinds of actions that characters may take.

In the following sections, I will explain all the various phenomena that are simulated over the course of a *Hennepin* simulation loop. Generally, the specific procedures that handled certain phenomena in my earlier engines, such as the procedures for handling births and deaths, are now handled by the action manager. That is, whereas top-down probabilities used to trigger certain critical events, now such events are reformulated as fine-grained character actions and occur through the basic functionality of the action manager (to be explained in Section 11.2.7).

11.2.3 Infrastructural Evolution

Whereas in my earlier systems I hardcoded specific procedures for evolving the *infrastructure* of a storyworld—*World*'s settlement activities and ship passages and *Talk of the Town*'s business operations—in *Hennepin* this evolution is handled by the general mechanism of character action selection. For instance, a road is built when: a character requests roadway access to her home, a county official processes the request, another county official plans the roadway, a county

place at any time. As such, the termination date is simply a matter of when the accumulation of actions constituting the history of a storyworld should end. Each podcast will recount events from different periods in a storyworld's history.

```
10204: [envy-spouse-of-love-interest] by Chester Winbush about George Brentlinger at Winbush residence (house)

10205: [admire-the-sunset] by Clyde Dolphin at Dolphin residence (house)

10206: [gaze-upon-the-stars] by Arthur Naab at a spot on public land

10207: [cry-over-past-action] by John Chilcote about make-fun-of at Chilcote residence (house)

10208: [have-a-drink] by Will Bracey at Wettengel residence (house)

10209: [hatch-revenge-scheme] by Harry Thore about John Stacy at Thore Rock Quarry

10210: [write-a-poem] by Edward Sibbett at Sibbett residence (house)

10211: [inspect-artifact] by Virginia Naab about posting for opening for a day laborer at Thore Rock Quarry at a campsite
```

Figure 11.1: An excerpt of *Hennepin* printout displayed during its procedure for world generation. The emergent events shown here express the variety of character actions that may occur in a *Hennepin* storyworld: some are contemplative, others are social, some occur within a character, and others involve the creation or examination of physical artifacts. Each is gated by *preconditions* that specify when and where and by whom and in what circumstances the action may be performed. In turn, each action changes the storyworld in some way upon being performed, according to its *effects*.

work crew travels to the planned location, and the work crew actually constructs the roadway segments. This procedure thus occurs emergently according to the coordination of a number of fine-grained character actions, each of which may dynamically incur various other effects, making roadway construction itself an emergent possibility space. This is how all character phenomena works in *Hennepin*, including the more benign phenomena that undergird the bureaucratic and infrastructural concerns of a county. As such, it is not really possible to describe how these phenomena are modeled in the way that I did for the earlier simulation engines. Instead, my description of character actions in Section 11.2.7 may be taken as providing this explanation in a general way.

11.2.4 Character Routines

Character routines work exactly like in *Talk of the Town*, except that characters with queued actions (about which more soon) are sent to the location where that action is to be taken, if that is specified. While I will explain more about action queues below, the general idea is that the action manager may schedule future actions for characters—for example, `attend-party`—and such scheduling may record a specific location at which the action must be taken. For example, if someone is to attend a character's birthday party at the latter's home on a specific date, the system handling character routines will automatically move the character to that location at the beginning of the corresponding timestep.[27]

Being There

Initially, I had characters take the action of actually going to a place where a scheduled action is be taken, but it led to thousands and thousands of boring actions constituted in characters heading to various locations at the beginning of a timestep. By this refined approach, characters simply begin a timestep at the places at which they need to be. To me, this evokes the crafting of scenes in narrative media, where no effort is taken to establish how everyone got to the location of that scene—it just starts, and everyone is already there. Unlike *Talk of the Town*, characters may move between locations during a timestep; this is handled by action effects (more on that below).

[27] In his feedback on an earlier draft of this thesis, Noah Wardrip-Fruin commented on the routine system: "I understand not modeling characters moving around the world. But it would be great if 'you won't believe what happened to me on the way here' was still possible" (personal communication, August 10, 2018). This is an interesting idea, and it could be accomplished by queueing transitional actions whereby a traveling character moves to intermediate locations in the county that are along the way to an ultimate destination.

11.2.5 Character Relationships

While *Hennepin* disposes of *Talk of the Town*'s abstract modeling of social interaction, the notions of *charge* and *spark* (see Section 9.2.5) are still inherited. As I will explain in the next section, however, updates to charge and spark values occur according to character actions. Specifically, a charge or spark update is now simply one kind of action effect, which means different actions will affect these affinities in different ways. As such, the system can explore the concrete history of a relationship, which was not possible in my earlier simulation engines. For example, the origins of an enmity could be reasoned about in terms of the specific character interactions that bred the contempt. Moreover, charge and spark values may be referenced in action preconditions, thereby gating certain kinds of actions depending on how characters feel about one another.

Discrete Relationships

Even though the evolution of charge and spark values works differently, discrete character relationships are still formed in the same way: as specified charge and spark thresholds are crossed, discrete relationships such as a `friend` or `enemy` are instantiated. Finally, *Hennepin* also still tracks a number of other social relationships, including: specific familial relations, such as `sister` or `cousin`; occupational relations, like `coworker` or `boss`; and residential connections, including `neighbor` and `former neighbor`.

11.2.6 Character Values

While it less clear whether or to what extent personality changes over time in the real world, *value systems* clearly change in the course of a life, sometimes by acute shifts. In *Hennepin*, l wanted to model this kind of change because I feel that it yields narrative potential—for instance, a character arc, transpiring over the course of the person's existence in a storyworld, could be undergirded in part by shifting beliefs about what is really important in life.[28] Moreover, if such change generally paralleled shifts in the (broadly construed) American value system that have occurred over course of this country's history, a basic verisimilitude would be imbued in *Hennepin*'s storyworld modeling. In this way, there are distinct appeals to modeling change over time both within characters and within a storyworld.

Planned Antecedents

To model such change, I decided to return to the *incrementation model* that I had implemented in *World*'s subsystem that simulates character knowledge phenomena, which was the subject of Section 7.3.16. In the discussion section of my paper on that work, I actually proposed that the approach could be extended to more concrete sociocultural considerations (citation inserted for clarity):

> This notion of using diegetically evolved vectors [...] suggests a more generalized approach that evolves sociocultural considerations beyond language. [...] Instead of operationalizing linguistic theories, this proposed integration could operationalize theories about general sociocultural transmission and evolution, e.g., the influential work of [Luigi Cavalli-Sforza and Marcus W. Feldman] [169]. [1037, p. 7]

Indeed, from the beginning, I had been interested in exploring the evolution of other kinds of sociocultural phenomena, but I never got around to it in *World*. One idea that Peter Mawhorter and I discussed at that time was to evolve vectors whose entries would map to specific cultural practices, for example, clothing preferences—in this way, a community's preference for pointy shoes might gradually evolve into a preference for curled shoes as bit flips propagated over time.

[28]In his feedback on an earlier draft of this thesis, Noah Wardrip-Fruin commented, "Reading the descriptions of extreme Hennepin values, I can already find narrative intrigue starting. If a character were portrayed as feeling that way, I would start wondering, 'How did they get that way?'" Indeed!

Later, in *Talk of the Town*, I became interested in the prospect of applying its modeling of character knowledge (beliefs about what is true) to sociocultural concerns (beliefs about what is important). In text that did not make the cut of our paper on that system [1061], I proposed an integration with the *Chimeria* system of Fox Harrell and collaborators (citation inserted for clarity):

> Here, let us consider *Chimeria*, an AI system that richly models social-group membership according to specifications of those groups that come from hand-authored *domain epistemologies* [454]. In an application of the system called *Chimeria: Gatekeeper*, the underlying domain epistemology defines the attributes that are prototypically associated with two races in a fantasy world. For instance, this epistemology specifies (among other things) that members of the first race like rough-spun garments, while members of the other race prefer fine clothing. As the system stands today, agents in *Chimeria* applications inherit from a single hand-authored domain epistemology that is static and cannot change according to events that may occur in the simulation. One can imagine, however, an integration of this system and ours by which the components of a domain epistemology are represented in our system as belief facets. By this integration, epistemologies would be instantiated at the level of the individual agent (which is more intuitive) and would be built up according to observed agent behavior in the simulation. That is, the component of a character's instantiated epistemology corresponding, e.g., to clothing preferences among members of some race in the game would get built up incrementally according to evidence that the character encounters as she goes about the world. Components of epistemologies could likewise propagate, deteriorate, and terminate just as knowledge does in *Talk of the Town*.

I never got to this work in *Talk of the Town* either, but finally in *Hennepin* I am exploring the evolution of sociocultural concerns for the first time. Here, I think that the resulting system is more interesting than *World*'s language system, because in this case the subject phenomenon (character value systems) is represented using a concrete representation scheme (values are strings), as opposed to the abstract language vectors that I utilized in the earlier work.

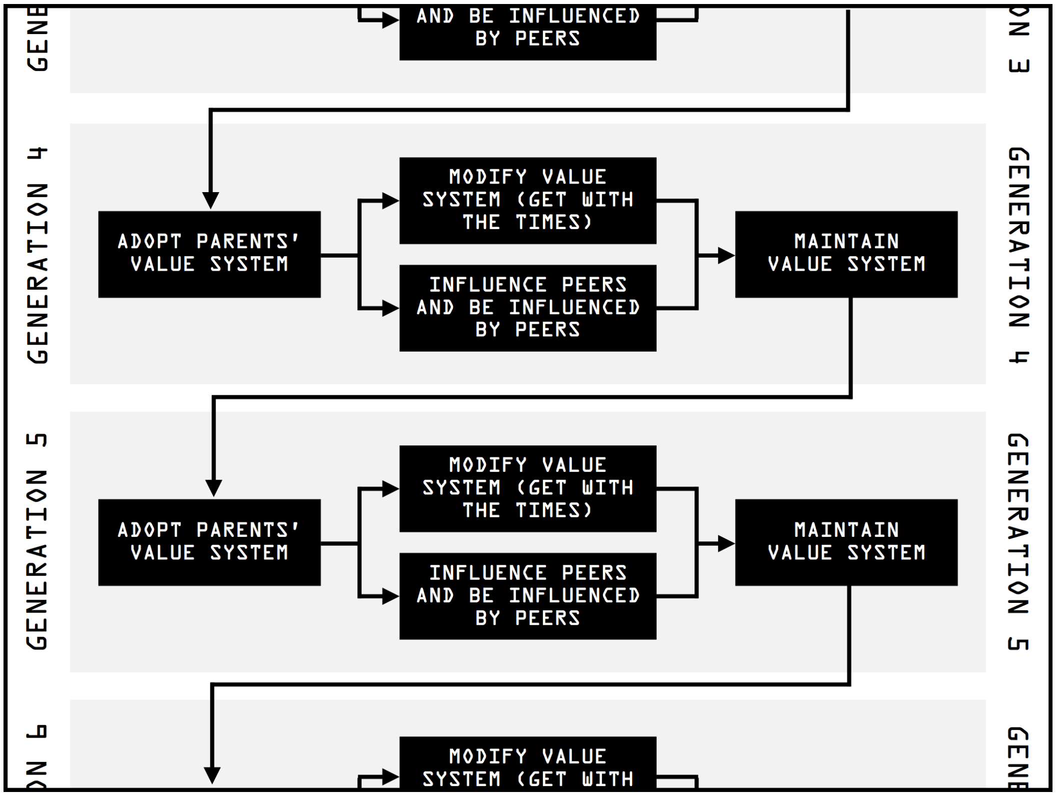


Figure 11.2: An illustration of the incrementation model that characterizes the evolution of *character value systems* in *Hennepin*. A character begins life by inheriting the value system of her parents, which she then proceeds to modify and adapt throughout her teens and twenties, a process that terminates as she reaches her early thirties and begins to simply maintain the modified value system, which she may then pass on to her children. In the period in which modification occurs, the character may be pulled toward anchor opinions that are meant represent the American zeitgeist of the current storyworld era (for example, diversity becomes more valued over time). Additionally, a character may be influenced by her peers or persuaded by an influential character who extols or denounces a value. Generally, this approach works according to the same method that was used in *World*'s modeling of character knowledge phenomena, which operationalized William Labov's *incrementation model* of language change.

Incrementation Model

In a *Hennepin* storyworld, character values change over time by an incrementation model that works similarly to the operationalization of William Labov's theory of language change that was utilized in *World*; this process is illustrated in Figure 11.2. Since I already explained the latter system at length in Section 7.3.16, I will not go into too much depth here. When a child is born in a *Hennepin* county, she inherits a value system from her parents in the same way that a personality profile is inherited: for each value, a parent is randomly selected, and the child takes after that parent by adopting the same value score, but with some mutation. Throughout her late teens and twenties, a character proceeds to modify her values over time under the influence of the historical era at hand, as well as the influence of her peers (particularly influential characters who may try to persuade her in some direction by performing certain actions). Finally, when a character reaches her early thirties, she stops updating her value system and instead maintains it, potentially passing it on to the next generation, per the incrementation model.

Zeitgeist Anchors

Rather than *World*'s random mutation (modeled as the flipping of arbitrary bits), here there is an external modulating force that guides the process. As the storyworld's historical era changes, a normal distribution modeling predominant character values shifts—this is modeled as the mean shifting, though the variance could also shift to generate different spreads. Specifically, for each of the thirty-four character values that were listed above, I authored definitions for normal distributions that are each associated with an era (represented as a date range). This authored procedural data captures that, for instance, in American history, at a macro level, `decorum` has become less valued over time while `diversity` has become more valued.[29] As time progresses, a new normal distribution takes hold,

[29] Here, I am simply being guided by my intuitions, since I do not see a point in actually doing research to operationalize a theory. The details of how this works are not especially

which has two effects. First, its mean acts as an anchor that draws the developing opinions of adolescents toward it—thus, if a teenager's evaluation of `diversity` is `3`, but the anchor is `13`, she may probabilistically adapt her belief to shift toward the anchor (thereby increasing it, in this case). Second, if a new adult is generated to move to the town, her value system is instantiated by sampling from each of the active normal distributions.

Persuasion

Additionally, there is another way that values can change: persuasion. As I will note at various points later on, characters may perform actions whereby a particular value is extolled or denounced. If a character who is still reformulating her value system is the recipient of such an action, she may shift her opinion of that value in the corresponding direction, depending on social concerns such as her affinity for the speaker.[30]

11.2.7 Character Actions

The fundamental unit of simulation in *Hennepin* is the *character action*. Everything that happens in a storyworld is constituted in discrete character actions that are gated by preconditions and that have associated effects that bring about changes to the storyworld. In this section, I will explain this system in detail.

---

important, since the narrative potential of this system—which is ultimately what I seek—will be produced simply by having character values change over time (both within a character and across a storyworld), especially as a result of concrete actions by influential characters.

[30]The phenomenon of persuasion is also modeled in *Dwarf Fortress* [17], as an entry in a fan-maintained Wiki explains: "Beliefs can be changed through successful arguments. In adventure mode, this can be used to change the core beliefs of your adventurer (and by proxy some of your needs). Since you can argue for beliefs you don't actually hold, you can convince a person of a belief you want to hold, and then get into another argument over said belief, and acquiesce to their position" [280, n.p.]. Someone should utilize this module in a scholarship simulation.

Action Definitions

An *action definition* is a collection of authored procedural content that defines a *prospective action* that may be taken by a *Hennepin* character at some time, at which point a concrete *action* of that type is instantiated. Each action definition contains specifications for the following components:

- **Name.** A unique name that captures the type of action that the definition specifies. Example names for actions that I have authored include `chit-chat`, `address-warmly`, `ridicule`, `have-a-drink-at-work`, `write-a-poem`, and `gaze-upon-the-stars`. The system does not reason over these names, which means they are meant only for human consumption. As these examples illustrate, the actions that I have authored are generally granular and tailored to specific social contexts.[31] To connect to a distinction that Richard Evans has spoken about [324], the granularity here is closer to *The Sims 3* [1207], which features actions such as `compliment-home` and `worry-about-relationship`, than to *The Sims* [792] or *The Sims 2* [793], whose actions are on the order of `talk` and `joke`.

- **Role definitions.** A list containing *role definitions* for all the *roles* that are associated with this action, one of which must be an *initiator* role. While an action is always initiated by a single character, additional characters and other entities may be incorporated into the action by being cast in one of its roles. For example, in the definition for `ridicule`, the following roles are defined: `insulter`, `insulted`, and `observer`. As I just mentioned, roles may be filled by entities beyond characters, such as homes, businesses, artifacts, values, or even other actions. For example, the action `gossip` has a role called `past-action`, which is filled by a past action that is the

[31] Condition logic enforces that actions are taken only in the appropriate contexts.

subject of a character's gossip. I will explain more about role definitions in the following section.

- **Preconditions.** A set of *preconditions* that must be satisfied in order for the action to be taken at some point. Preconditions are evaluated with respect to a candidate *binding*, which specifies a set of candidate entities to be cast in an action's roles. For example, a candidate binding for `ridicule` might look like this: `{insulter: Meehan, insulted: Klein, observer: Schank, observer: Dehn}`. A precondition, then, specifies constraints on who may be cast in what roles at what times. As an example of a `ridicule` precondition, `insulter.personality.cruel >= 25` requires that a character must be particularly cruel in order to ridicule someone. In this case, the *action manager*, described in Section 11.2.7, would evaluate this precondition by using the candidate binding, where the character `Meehan` is bound to the role `insulter`. Another precondition might depend on the relationship between the insulter and insulted, for example, `insulter.charge(insulted) < 0`, in which case the character `Klein` would be bound to the role `insulted` to evaluate the condition. More broadly, a precondition may specify anything about any modeled aspect of the storyworld. At a technical level, my authored preconditions are Python *lambda expressions* [1290, p. 58] that have access to both the candidate binding and the storyworld state.[32] Here is an actual code snippet defining a precondition for the action `cringe-over-embarrassing-memory` (`c` is bound to a data structure for the candidate binding in which individual

[32]Lambda expressions are statements in the *lambda calculus*, a formalism that was introduced by Alonzo Church in a 1932 paper titled "A Set of Postulates on the Foundation of Logic" [187, 163]. Among Church's students were Alan Turing and Stephen Cole Kleene, the latter of whom advised Robert Constable, who advised Joe Bates, who coadvised Michael Mateas, who coadvised me.

role bindings may be accessed by dot notation):

```
lambda c: c.cringer.associates(
    action=c.past_action, tag='embarrassment'
)
```

Lastly, I would like to note an authoring pattern whereby a precondition is simply a specified probability, which allows for an author to turn down the chance of an action occurring even when all its binary preconditions hold. For example, take this precondition:

```
lambda c: random.random() < 0.5
```

With this condition in place, an action that would otherwise be taken has only a 50% chance of being taken. This can be used to turn down the frequency of actions that should occur less often.

- **Effect definitions.** If an action is taken, a set of associated *effects* may be executed to update the state of the storyworld; in this way, actions change the world. In the action definition, an author specifies a series of *effect definitions* that couple *triggers* and *effects*. A trigger is a condition that specifies the contexts in which an effect may be executed, and the effect specifies how exactly the storyworld will be changed if the rule fires. As such, action effects in *Hennepin* are *conditional*, which means they are not strictly executed when an action occurs, but instead the effect trigger is evaluated, and only if those conditions hold are the actual effects executed. The trigger conditions work just like preconditions, which means they may specify anything about the storyworld and operate over an action's bindings.[33] Like the conditions,

[33] To be precise, action preconditions operate over candidate bindings, while effect triggers operate over instantiated bindings, since by then the action has already been taken.

effects are defined as Python lambda expressions, but instead of returning `True` or `False`, they call functions or methods whose execution changes the world. Specifically, an effect is defined as a dictionary that couples the trigger conditions and the actual effect. Note that such structures are similar to *production rules*, like the ones that are used in Sheldon Klein's murder-mystery generator and *Comme il Faut*, as I discussed in Section 4.1.3. Here is a code snippet that defines an effect of `ridicule` that causes the character who is ridiculed to lower her charge toward the person who has just ridiculed her:

```
{
    "conditions": lambda c: c.me is c.insulted,
    "effect": lambda c: c.me.update_charge(
        c.insulter, -5
    )
}
```

One authoring pattern is to conditionalize an effect on a character's personality profile, which means the same action will affect characters differently, depending on variation across their personalities. This works in the style of Richard Evans's modeling of personality traits as conditionals—for example, a `thrill seeking` trait being represented by a rule of the form *If I perform a risky action* → *Excitement* [325, p. 37]. Here is an example effect definition that follows this pattern to specify that cruel characters are impressed by cruel acts (and thus increase charge toward the culprits):

```
{
    "conditions": lambda c: (
        c.me is not c.insulted and
        c.me.personality.cruelty >= 25
    ),
    "effect": lambda c: c.me.update_charge(
```

```
            c.insulter, 5
        )
    }
```

- **Salience definitions.** Each action that a character knows about will have an associated *salience* value that captures how notable the action is to that person. Salience affects the likelihood of remembering an action, propagating knowledge about an action, and taking actions that cast past actions in their roles. For example, a character is more likely to remember giving birth to a child than drawing a picture in school, and this is captured by the former having a higher salience. Likewise, characters are more likely to gossip about an attempted arson than a character admiring the sunset. Moreover, it would probably not even make sense to gossip about someone admiring the sunset, because its salience is low—to capture this, the `gossip` action definition might say that its `past-action` role can only cast past actions whose saliences exceed some threshold.[34] Critically, action saliences are also conditional, which captures the intuition that an action may be more notable to some characters than others. For example, the salience definitions for the action `give-birth` specify that while the action will be fairly salient to anyone who hears about it, it will be *extremely* salient to the mother who gave birth. Salience definitions resemble the production rules specified in effect definitions, except that that the right-hand side specifies a *salience increment*, rather than an effect. Here are two salience rules that capture that a `ridicule` action is more notable to the character being insulted than to the one doing the insulting:

[34]This would be captured by a precondition along the lines of `gossiper.salience(past_action) > 50`).

```
[
    {
        "conditions": lambda c: c.me is c.insulted,
        "value": 750
    },
    {
        "conditions": lambda c: c.me is c.insulter,
        "value": 200
    }
]
```

Here, the `value` component specifies a salience increment on an arbitrary scale. As I will explain below, when a character experiences or hears about an action, its salience to her is incremented according to all of the salience rules that fire. In this way, if an event becomes the talk of the town (county), it will become more salient to everyone simply because they will keep hearing about it. Action saliences decay from the passing of time, and forgetting happens when a low salience threshold is crossed. To be clear, saliences are specifically attached to the knols that collect knowledge about an action.

- **Special.** Certain prospective actions are not included in the pool that is used by the action manager for typical action selection—these are called *special actions*. Generally, special actions are contingent on earlier actions having already been taken, which means it would not make sense for a character to spontaneously take one. For example, the special action `construct-frame-for-building` can only be taken if the action `lay-foundation-for-building` was already taken. It may seem like this contingency relation could simply be expressed in the precondition logic for `construct-frame-for-building`, but this would not work because the same action bindings must be used for `lay-foundation-for-building`. To solve this problem, I have developed an authoring pattern whereby the effects of one action may cause a special action to be *queued* with a set of prespecified bindings. For example, one effect of `construct-frame-for-building` is that a special action is queued for the builder, with prespecified bindings that cast the same building in the `building` role (so that the character does not spontaneously decide to build a frame on some other foundation laid by another character). I will explain exactly how action queueing works below.

- **Cooldown.** Finally, some actions should not be taken multiple times during some duration, and the system is able to enforce such concerns by utilizing the *cooldown* component of an action definition. This component is specified as the number of timesteps until the action may be taken again.[35] Generally, I utilize this authorial affordance to combat repetitive action, though it could also be to enforce normative social behavior. An example value that I have authored is a 30-timestep cooldown on the action `gaze-upon-the-stars`, which I specified because characters were constantly taking the action, even multiple times on the same timestep.

Role Definitions

As I noted in the last section, character actions are associated with *roles* that are filled by characters or other entities in a storyworld. As such, a critical component of an action definition is a collection of *role definitions* that define all of the roles that may be cast in that action.

In the opening of this chapter, I mentioned that *Hennepin* is deeply inspired by two earlier systems that were developed in the Expressive Intelligence Studio:

[35] More precisely, a cooldown declaration for an action will have an associated set of bindings, which means the initiator cannot take that action again *with the same bindings* until the cooldown is lifted. More precisely still, the associated bindings will be a set of *minimal bindings*, meaning *optional roles* are not included. I will explain what these concepts mean below, but this is not a very important detail.

Comme il Faut (*CiF*) [804], which is the AI system that undergirds *Prom Week* [799, 803, 1085], and *Ensemble* [1089], a successor to *CiF* that carries on this *social-physics tradition*.[36] In *CiF*, there is an implicit notion of a role in that each action (called a *social exchange*) has an *initiator* and a *responder*, and some actions may also incorporate or otherwise reference a third character, called the *other*. *Ensemble* still requires an initiator and a responder, but it extends *CiF* by allowing for any number of additional characters [1089, p. 25–26]. These additional characters do not directly participate in the social exchange, but they may be referenced in its preconditions and effects.

With *Hennepin*, I sought to extend this development further to make an even more flexible role schema. First, while work in this tradition specifically concerns social interactions (naturally), I am also interested in modeling the internal lives of characters, as I have expressed throughout this dissertation. As such, my schema does not require an action to have a recipient (as with

[36]The social-physics tradition centers on the notion of character 'social exchanges', which are patterns of social interaction that are defined by authored procedural content. In *CiF*, social exchanges were originally called 'social games', but the name was changed as that term came to be associated with online games that are integrated into social platforms [623]. In the context of this tradition, the term was actually first used by Michael Mateas and Andrew Stern in their work on *Façade*. In that project, 'social game' refers to a proceduralized pattern of social interaction whereby the player performs *discourse acts* that modulate a numeric score that is associated with the social game [778, pp. 94]. For example, the *affinity game* tracks whose side the player is taking—Grace's or Trip's—by maintaining a score that is modulated as the player performs discourse acts over the course of gameplay. The term is actually adapted from Eric Berne's *transactional analysis*, a framework that applied game theory to social psychology, resulting in the identification of patterns of human interaction that Berne calls *games* [104]. When Josh McCoy began working with Michael Mateas, after the latter arrived at UC Santa Cruz from Georgia Tech, they refined the notion of a social game under the inspiration of Erving Goffman's *dramaturgical analysis* [399], which McCoy had already studied as an undergraduate at Earlham College [797]. This initial reformulation is the subject of the first two papers on what would become the *CiF* project [798, 802], and its subsequent development spans from two other early papers [800, 801] to the solidified notion as it is recounted in an extensive series of later works [806, 799, 803, 770, 804, 1085], including the *Prom Week* core team's respective dissertations [807, 1264, 1084, 984]. Finally, more recent offshoots of this work include the *Immerse* project [1140, 1139, 335] and the aforementioned *Ensemble* project, which itself splintered into separate initiatives that were informally (but affectionately) referred to as '*Ensemble* west' [1089], based at UC Santa Cruz, and '*Ensemble* east' [1267, 1268], based at American University (where Mike Treanor, Josh McCoy, and Anne Sullivan were each based for a period).

the 'responder' role of *CiF* and *Ensemble*). For instance, the aforementioned `cringe-over-embarrassing-memory` occurs within a character, and as such it has no recipient. Next, in the tradition of *Talk of the Town*'s simulation of character knowledge phenomena, I wanted actions to be *observable* by other characters, and moreover I wanted actions to have potentially many participants. As such, my schema allows for action definitions that incorporate an arbitrary number of active, physically present participants. Finally, I wanted my schema to support actions that incorporate entities that are not characters—for instance, in `cringe-over-embarrassing-memory`, a past action is the subject of embarrassment, while `write-a-poem` involves a poem artifact that is produced by the action. In the end, my desire for an extremely flexible role schema is what caused me to abandon my plan to integrate *Talk of the Town* and *Ensemble*, a decision that led to the development of this entirely new simulation engine.

In *Hennepin*, a role is defined compositionally according to its attributes, and these attributes drive and constrain the action manager's procedure for selecting candidate action bindings during action selection (as I will explain shortly). The following attributes are at an author's disposal in defining an action role:

- **Action.** This is not an attribute, but rather the particular action definition that the role is associated with. Roles are defined within an action definition, and thus with regard to a specific *action*.

- **Name.** This is likewise not an attribute, but instead a *name* that is attached to a role definition for human legibility. Example names include `admirer`, `hugger`, and `insulter`.

- **Initiator.** When this attribute is present, an entity that is cast in this role will serve as the *initiator* of the corresponding action. This notion is inherited from the social-physics tradition. When the action manager is handling action selection for a given character, it is specifically reasoning over actions for which that character may be cast in an initiator role.

- **Recipient.** A role with this attribute casts a *recipient* of the action. Like the 'initiator' attribute, this one is also inherited from the social-physics tradition, though I have renamed 'responder' according to my preference for this term.[37] There is nothing particularly special about a recipient role, except that the system records that an entity cast in one is a recipient, which can be an important type of information at the curation stage.

- **Bystander.** This attribute specifies that the role is cast by a *bystander* to an action. Bystanders are cast probabilistically from a pool of nearby characters, and they may witness actions (to form memories about them) and be referenced in action effects, but they do not actively participate in the action. For instance, a bystander who witnesses a character ridiculing someone else may lower their charge for the offending party (or raise it if the offended party is an enemy, or whatever else an author may wish to specify).

- **Hearer.** A role with the special *hearer* attribute is cast by a character who learns about an action after the fact. This attribute exists to allow for an author to specify different effects for people who hear about something secondhand. For instance, a character who only hears about a `ridicule` action may be less repulsed than one who witnesses it firsthand.

- **Character.** This attribute specifies whether a character must be cast in

[37] In the social-physics tradition, character actions are 'social exchanges' in which an initiating character proposes an action that a recipient either accepts or rejects, and this decision determines which effects will be triggered. As such, in this work the term 'responder' makes sense, but in my scheme the recipient of an action does not necessarily actively participate in it.

this role, which guides the action manager's formation of the entity pool that is sampled to form candidate action bindings.

- **Action.** If this attribute is present, the role can only be filled by another action. Again, this guides the action manager's binding process.

- **Location.** This specifies that a location (county, parcel, lot, building, etc.) must be cast in the role.

- **Value.** A value must be cast in the case of this attribute being present. Example actions that cast values include `extol-value`, `denounce-value`, `agree-about-value`, and `disagree-about-value`. For instance, two characters with conflicting extreme opinions on the value `tradition` might participate in the latter action.

- **Artifact.** Some actions have roles that cast artifacts, such as `examine-artifact`, which characters use to examine an existing artifact, and `draw-a-picture-of-action`, whereby a character creates a new artifact (a picture) that recounts an action (for example, a house burning down). The latter action extends my exploration in *Talk of the Town* of artifacts that transmit knowledge—this was explained back in Section 9.2.11—in that by examining that picture, a character would learn about the house burning down.

- **Symbol.** It can sometimes be useful to define roles that should be filled by arbitrary types, such as a string or a number. For example, in the action `come-up-with-name-for-town`, the proposed name for a new town is simply a string, and so the corresponding role would have this *symbol* attribute. In such a case, however, the action manager will have no way of knowing how to construct a pool of candidate bindings, which means such a pool needs to be explicitly defined in the action definition. This is done by specifying a *candidate-pool directive*, which is defined as arbitrary code that returns a list of objects, which is then treated as the pool of candidates for that role. For instance, here is the candidate-pool directive for the action `berate-employee` (this lambda expression takes the initiating character as its argument `i`, since it precedes there being any candidate binding):

```
"binding pool directives": {
    "employee": lambda i: [
        p for p in i.location.people_here_now if
        p.routine.occasion("work")
    ]
}
```

- **Absent.** This attribute specifies whether an entity cast in the role must be physically present at the location of the action; this guides the action manager's formation of a candidate pool for bindings. For example, in the action `talk-behind-someones-back`, the character who is being gossiped about should not be physically present, in which case this attribute would be included in the corresponding role definition.

- **Build.** If this attribute is activated, the role will be cast by something that does not exist yet. This is used in cases where an action produces something new—for example, in the action `write-a-poem`, a poem artifact is produced by the action, and the corresponding role definition makes use of this attribute. While usually such roles will be cast by newly created artifacts, that is not always the case. In the action `give-birth`, for instance, a new character is created to fill the associated `newborn` role. In order for the action manager to actually construct the appropriate entity, the action definition must contain an associated *build recipe*. This is defined as arbitrary code that returns an object, which is then cast in the role.

- **Minimum number cast.** An author may specify that a role is optional in the sense that an action can still occur without any entity being cast in the role. This is defined by setting this attribute to 0, which makes for an *optional role*. In turn, it may also be set to other values in cases where that makes sense. For example, the action `make-the-whole-group-laugh` requires at least four recipients to be cast, since otherwise a group is not being made to laugh.

- **Maximum number cast.** Likewise, an author may specify a *maximum* number of entities to be cast in a role. Typically, I define bystander roles to have a fairly low number, since this saves on computation and it is not believable, for instance, to have dozens of bystanders looking upon an innocuous action like `chit-chat`.

- **Chance.** Once the specified minimum number of entities have already been cast (but before the maximum number has been met), a *chance* value specifies the likelihood of casting additional entities in that role. This essentially models a notion of how likely an event is to be witnessed—for example, the chance would be much higher for a bystander role of the action `start-fight`, as opposed to `chit-chat`.

Action Selection

In *Hennepin*, a module called the *action manager* carries out action selection, and it does so according to a simple operating principal: to maintain believability, characters need only to *refrain* from taking actions they would *not* believably take, which means it is rarely necessary to take the most fitting action. In this way, with regard to believability, the system attempts to produce *satisficing*, as opposed to *optimal*, character activity. This distinction originates with Herb Simon's seminal

1956 paper “Rational Choice and the Structure of the Environment”:

> The central problem of this paper has been to construct a simple mechanism of choice that would suffice for the behavior of an organism confronted with multiple goals. Since the organism, like those of the real world, has neither the senses nor the wits to discover an “optimal” path—even assuming the concept of optimal to be clearly defined—we are concerned only with finding a choice mechanism that will lead it to pursue a “satisficing” path, a path that will permit satisfaction at some specified level of all of its needs. [1160, p. 136]

In the case of this project, the action manager has one need: produce believable character activity. This principal specifically manifests in a policy for action selection whereby: prospective actions are randomly targeted, candidate bindings for those actions are randomly assembled, and if the preconditions hold for any such candidate binding, the action is taken immediately.

Though I figured that it would hinder believability, I tried this approach as a first attempt because I was worried about the computational burden of having hundreds of characters decide between potentially thousands of actions—and potentially millions of bindings—thousands of times over the course of a storyworld’s history.[38] As I have mentioned above, I envision such a history resulting in an accumulation comprising on the order of a few million character actions. Due to the size of a *Hennepin* storyworld (in terms of the number of entities that may be cast in action roles), even selecting an optimal binding for a single predetermined action could take a very long time.

In the social-physics tradition, during *volition formation*, the system selects the most optimal action by evaluating thousands of social rules for every possible action and every possible binding. This approach is feasible when the storyworld

[38]Another system that simulates on the order of thousands of characters is the recent indie game *Project Highrise*. Though it models characters in less detail than I do here, the game’s simulation works in real time, and so its creator Rob Zubek also makes concessions to work under competing design and computational constraints, as he explains in a book chapter titled “1000 NPCs at 60 FPS” [1384].

contains only a dozen or so characters—as in *Prom Week* [799, 803], *Mismanor* [1210, 1211, 1212], and the kind of experiences that are envisioned in the *Ensemble* project, such as Ben Samuel's *Writing Buddy* [1086] and recent prototypes developed by Mike Treanor and collaborators [1268, 1088]—but it is probably not feasible in a *Hennepin* storyworld. While I knew I would have to take a satisficing approach, I figured that randomly selecting viable actions would produce strange character activity—to my surprise and good fortune, this was not the case.

Through testing, both by myself and with others, I found that believability did not seem to be hindered as long as characters did not take actions that were totally unreasonable in the circumstances. Given that this is what action preconditions specify—their purpose is to outright preclude actions that characters should not take—believability was thus not hindered. Even if a more fitting action is available but not taken, humans will tend to unconsciously narrate reasons for why the character did what they did (and did not do what they did not do)—thus, by targeting this tendency, I had utilized the design pattern of *apophenia hacking*, a notion that I introduced in Section 9.3.[39] I am glad that I tried the most basic approach first, since otherwise I would have likely crafted up a selection policy that would do more reasoning (and thus require a lot more computation). This is an example supporting the idea that the simplest approach should always be tried first, even if it seems unreasonable, because such an approach will provide the greatest payoff if it does happen to work [37].

Here is a very high-level sketch of the action manager's procedure for action selection, which I will describe in more detail momentarily:

```
randomly select character from actors pool
    if character has (semi)urgent queued action, target
        if successful, execute action and move on
```

[39]Indeed, this move appears to be in line with the apophenia hacking that Matt Brown has promulgated in the context of *The Sims 2* [146], which I noted above.

```
otherwise, target randomly selected action
evaluate initiator preconditions
randomly assemble minimal bindings
    evaluate required-role preconditions
    for each optional role
        randomly select binding
        evaluate optional-role preconditions
    if successful, execute action and move on
```

On each timestep that will be simulated, every character in the county is added into an *actors pool*. Next, the action manager randomly selects a character from the pool and attempts to settle on bindings for an action that the character will take as an initiator. First, the manager checks whether the character already has a (semi)urgent action queued up, a notion that I will explain shortly. If she does have an action queued up, the manager targets it with the specified bindings—if this is successful (all preconditions hold), the character takes that action and the manager selects a new character from the actors pool.[40] Assuming the character does not have a queued action, the manager randomly selects a prospective action from the pool of generic actions (i.e., special actions are not considered). Next, the manager evaluates the initiator preconditions for that prospective action.

While I did not mention this above, the preconditions in an action definition are actually split into three groups: *initiator preconditions*, *required-role preconditions*, and *optional-role preconditions*. This separation is meant to reduce computational burden, since it results in preconditions being evaluated only as they become relevant in the process of assembling a set of candidate bindings. The initiator preconditions do not have access to the rest of the candidate bindings (because they have not yet been assembled), which means they only specify what must hold for the initiator and the storyworld at large. Most often, initiator preconditions operate on a character's personality profile, so that certain actions

[40]If there are multiple queued actions, the one with the highest priority is targeted.

may only be taken by characters with certain profiles, which results in character actions that express the personalities of their initiators.

If the initiator preconditions hold, the action manager proceeds to randomly assemble a candidate set of *minimal bindings*—that is, bindings for only the required roles in the action definition. A required role is one whose specified minimum number to be cast is greater than zero (excluding roles with a `build` attribute). To randomly assemble a binding, the manager first builds a *candidate pool* for each of the required roles by operating over the definitions for those roles. For instance, if a role definition includes the `character` attribute but lacks the `absent` attribute, then the manager will add all the nearby characters (co-located with the initiator) into the pool. Alternatively, if the action definition includes a candidate-pool directive for that role, that is used instead to construct the pool.

Once all of the pools have been selected, the ordering of their elements is shuffled (to prevent repetitive binding) and a *Cartesian product* of the pools is constructed, which will contain every possible minimal binding.[41] From here, the system attempts to target each minimal binding one by one, in each case evaluating them against the action definition's required-role preconditions. Required-role preconditions operate only over the bindings of required roles in an action definition. If these preconditions hold for any minimal binding, the action will be executed, but first the manager attempts to also fill the optional roles.

To cast entities in the optional roles included an action definition, the system again constructs candidate pools for each (again ignoring roles with a `build` attribute). Now, instead of forming a Cartesian product of the pools, the manager simply attempts to fill each optional role one at a time (until the maximum number to be cast in that role is reached, or the pool runs out). In attempting to

[41]To be clear, this is not actually constructed and stored in memory, but rather each instance in the product is generated in a lazy manner.

cast an entity into one of these optional roles, the manager evaluates the action definition's optional-role preconditions, which may operate over an entire set of candidate bindings. Once each optional role has been cast or all the pools have been exhausted, optional role binding has been completed.

At this point, the manager has successfully assembled a viable set of bindings, and so the action may be taken by the initiating character. Had every set of minimal bindings violated the required-role preconditions, the action manager would have *backtracked* [1036, p. 84] to move on to another candidate action, again by randomly selecting from a pool containing all the generic actions (minus any for which targeting has already failed). Alternatively, a prospective action will also be abandoned if the initiator preconditions do not hold. If the action pool is exhausted for some character, she will not take any action at this time, and in turn she is removed from the actors pool. When this happens, it means that every set of minimal bindings was considered, but none of them were viable.

Once an action and bindings has been selected for a character, she takes that action, which leads to a concrete realization of it and the execution of a number of effects, as I will explain in the next section. Depending on a few concerns, the character may or may not be removed from the actors pool. First, if the character has a queued urgent action, she will remain in the actors pool. If this is not the case, a probabilistic trigger determines whether the character will remain in the pool—this probability depends on the character's `activity_level` personality trait, which makes more active characters act more often. An authorial affordance of this procedure enables the designation of some characters as being more important than others, which could be modeled by simply turning up the likelihood of remaining in the actors pool for the central characters. In this way, a subset of characters would act more often, and the rest of the characters in a county would constitute a kind of backdrop. I do not take this measure because I want every character to be important during the simulation, though I do plan to do such homing at the level of curation, in *Sheldon County*.

Concrete Actions

Once a prospective action and bindings have been selected for an initiating character, the action is taken and a representation of it (as a past action) is stored in the chronicle. The following kinds of data are stored to represent a past action for reasoning later on (both during the simulation and curation phase):

- **Number.** As each action occurs, it is attributed am *action number* corresponding to its position in the chronological order of all actions. This captures the ground truth of the temporal ordering of simulated events, which is obviously useful information in the curation phase.

- **Name.** This is inherited from the name field of the corresponding action definition, which allows for reasoning about an action's type.

- **Timestep.** The timestep on which the action occurred.

- **Location.** The location at which an action occurred.

- **Causes.** The earlier actions that caused this action to occur, by contingently unlocking it or explicitly triggering it. Contingent unlocking may be identified in a few different ways. First, when an action binds a role with the `action` attribute, the action bound to that role can be recorded as having caused the one at hand. For a given action, each such bound action is added to this list. Additionally, if an action explicitly triggered this one, through an effect of the former causing the latter to be queued, it will also be added into the list (more on this soon). This is how causal bookkeeping (see Section 5.4) works in *Hennepin*—though subtle, this is the most important aspect of the entire simulation engine, since it greatly facilitates story sifting in *Sheldon County*.[42]

- **Caused.** Subsequent actions that were caused by this action, either due to contingent unlocking or explicit triggering. As an action adds past actions into its `causes` record, it is in turn added into the `caused` listings associated with those past actions. This is the other half of causal bookkeeping.

- **Bindings.** All of an action's bindings, with each entry coupling a role and the entity cast in that role.

- **Initiator.** The character who took this action, that is, the entity bound to the role with an `initiator` attribute. An action can only have one initiator in *Hennepin*, as a simplifying measure. I could specify a new role attribute for a secondary initiator, but I have not found a use case for that as of yet. If I do, however, it will be easy to specify—by simply adding a new role attribute—and this is a core appeal of my compositional role definitions.[43]

- **Recipients.** The recipients of an action are all of the characters who are

[42]As Ian Horswill noted in his feedback on an earlier draft of this thesis, Stacy Marsella and Jonathan Gratch's *Ema* project employs a similar mechanism for what I call causal bookkeeping [415]. In this system, an agent generates causal interpretations for event sequences in order to produce corresponding emotions in a way that operationalizes *appraisal theory* [1111]. For instance, in light of an unfortunate event, an agent will trace backward along the causal chain preceding the event to assign blame; if the agent itself to blame, it "feels" guilt, but if someone else is to blame, it responds with anger. One difference here is that causal bookkeeping in this system is subjective: agents themselves carry out the procedure by operating over beliefs about the world and other considerations, including the intentions of other agents [415, p. 75]. In my approach, causal bookkeeping is carried out by an omniscient agent, and as such the causal records produced thereby reflect a ground truth for the simulated world. Nonetheless, I find the subjective treatment of the process to be quite interesting.

[43]Indeed, the individual attributes in the schema were added over time as new ones became needed or desired.

bound to roles that have the `recipient` attribute. As such, an action can have multiple recipients; this is another departure from *Ensemble*.

- **Bystanders.** The bystanders to an action are all the characters who are bound to roles that have the `bystander` attribute. As I noted above, bystanders to an action witness it, but do not actively participate in it. When someone is a bystander, everyone else involved in the action knows it. I could also specify a new role attribute indicating a secret observer, if I encountered a use case for doing so.

- **Nucleus.** The initiator and the recipients of an action compose its *nucleus*, which are the active participants in an action. Sometimes it is useful to isolate these particular entities.

- **Participants.** An action's *participants* comprise its initiator, recipients, and bystanders, but not any absent characters or non-character entities. To be clear, these are not the active participants in an action, but rather all of the characters who were present and participated in any way (including simply witnessing).

Building Entities

If any of an action's roles have the `build` attribute, the action manager follows the associated build recipes to construct the entities that will be cast in that role. For example, in the action `write-a-poem`, a poem artifact is built, while in `give-birth`, a new character (the newborn) is generated. When such an entity is built, it is added into the action's bindings accordingly.

Action Effects

Once a concrete action has been instantiated, its effects are conditionally *executed*. As I explained above, effects are defined as production rules that couple conditions (on the left side) and arbitrary code that will be executed if the conditions hold (on the right side). To trigger these rules, the action manager takes every participant in the action (including bystanders) and evaluates each rule—if a rule is triggered, the code capturing its actual effect is executed. Additionally, the action manager does this for any actions that are *relayed* by the action that has just transpired. For example, when a `gossip` action casts a past action in its `past-action` role (this being the past action being imparted of the gossip), the effects of that past action (which is being relayed by virtue of the `gossip` action) will also be triggered. This could cause someone hearing about a `ridicule` action to lower their charge toward the offending party, for instance. Critically, if the character already knows about the relayed action, its effects will not be executed again.[44] One critical authoring pattern is to define an effect whereby code is executed that leads to the *queueing* of a subsequent action as a kind of explicit triggering of a potential subsequent action by this one; I will discuss this below.

Memories, Knowledge Propagation, and Salience Updates

In Section 11.2.9, I will describe how actions trigger updates to character knowledge, but for now I will briefly summarize. When an action occurs, its

[44]This was not in place originally, and it led to the effects of some actions being far too intense. For instance, if a character was `ridiculed` by someone else, each time they thought of the action or heard about it later on, the effect of it would be executed just as it had when it was first experienced, which meant the insulted character's charge for the insulter would drop to extremely negative values. Ideally, the system would decay the strength of an effect each time it is executed, but there is unfortunately no general way of modeling this, since effects are defined as arbitrary code. I thought about defining secondary and tertiary effects for cases of an action's effects being triggered for the second and third time, but I decided that the authorial burden would be too immense.

participants and other witnesses form a memory about the action. Later, this knowledge may be propagated by performing actions that reference the earlier action. Likewise, characters may perform actions that create artifacts that encode knowledge about earlier actions—for example, `draw-a-picture-of-action`—and by examining such artifacts characters learn about the events they recount. Finally, learning about an action, or hearing or thinking about one that is already known, boosts its salience to the beholder.

Fulfilling Aspirations

As I will explain in more detail in Section 11.2.8, actions may cause character aspirations to be fulfilled. Whenever an action occurs, the *aspiration manager* uses it to evaluate every aspiration trigger for all aspirations that are held by living characters; if a given one fires, the corresponding aspiration has been fulfilled. For instance, if a character aspires to become a police officer, a `hire` action that represents the hiring for that position would fulfill the aspiration.

Action Queues

Perhaps the most critical aspect of *Hennepin* is its modeling of actions that explicitly enable the potential emergence of future actions—this is handled by a mechanism for *queueing* future actions along with preassembled bindings. That is, each character has an associated *action queue* that indexes all her queued actions. While it would be possible to simply rely on the precondition logic in action definitions to handle the unlocking of future actions according to past ones, it is easier from an authorial standpoint (and more efficient from a computational standpoint) to store the causal potential as the earlier action transpires.

For example, if a character takes the action `lay-foundation-for-building`, this contingently unlocks a subsequent `construct-frame-for-building` action. This could be modeled by including a precondition in the action definition for `construct-frame-for-building` that requires that an earlier `lay-foundation-for-building` occurred, but then this precondition will be evaluated constantly as characters who have no business constructing a frame (because they have not already laid a foundation) attempt to take the action (due to the action manager's policy of random targeting). Moreover, the `construct-frame-for-building` action should maintain bindings from the earlier `lay-foundation-for-building`, namely the entity bound to the `building` role. This could be accomplished in tandem with the approach of using condition logic by specifying a candidate-pool directive that says to look at the `building` binding for an earlier `lay-foundation-for-building` action, but this would mean retrieving information that could have easily been stored when the earlier action transpired—especially considering that the earlier action was almost certain to lead to the subsequent one.

Another strong use case for queueing actions pertains to character reactions that should occur immediately after a precipitating action transpires. For instance, if a character performs the action `start-fight`, then either the initiator or recipient should do something next in response. If such a response were not queued, the initiator might be removed from the actors pool, or worse, the action manager may randomly select an inappropriate action for one of the participants to take next (for instance, `gaze-upon-the-stars`).[45]

Subsequent actions are queued as part of the execution of the effects of the precipitating action. Specifically, a method of the action manager is called that requires information of the following type:

- **Trigger.** The action that will have precipitated the queued action (if it

[45] Actually, I kind of like the image of a fight being instigated and one of its participants responding by gazing upward contemplatively.

does ultimately occur) is called its *trigger*. When a queued action is taken, the trigger will be automatically added into the concrete action's `causes` record. This is causal bookkeeping, discussed at length in Section 5.4.

- **Action name.** The name of the queued action.

- **Bindings.** Partial or complete bindings for the queued action; if partial bindings are supplied, the action manager will attempt to bind the additional ones when the queued action is eventually targeted. One special kind of binding concerns a *null initiator*, whereby an action is queued that is not actually taken by a character. An example such action is `burn-to-the-ground`, which is queued when a building is set on fire. When such an action is queued, the null initiator is added into the actors pool, and the actions may be targeted in the case that the null initiator is selected to act. When the null initiator has no actions queued, it does not appear in the actor pool. This is a similar concept to Chris Crawford's notion of *Fate* being a character [224, pp. 209–226].[46]

- **Urgent.** An *urgent* queued action will be taken on the current timestep, and a character who queues an urgent action will be kept in the actors pool (or added back into it). If an urgent action cannot be taken (due to its preconditions not holding or to the remaining roles not having viable bindings), that is an authoring error. An action should never be queued as urgent if it cannot be immediately taken, since that it is the operational meaning of 'urgent' in this sense.

- **Semiurgent.** Whenever it is a character's turn to act, the action manager will always target any *semiurgent* actions in her queue, but having one in

[46]In *The Sims 3* [1207], towns are interestingly modeled to form desires (for instance, pertaining to gender ratio) and take actions in pursuit of them [324].

the queue does not force the character back into the actor pool (as with urgent queued actions). An example semiurgent action is `break-up-fight`: if a character decides to break up a fight, she should attempt to do so on her next chance to act this timestep, but it is fine if the timestep ends without her getting a chance to do so. To be clear, the distinction between urgent, semiurgent, and nonurgent queued actions is as follows:

- *Urgent.* If an urgent action is in a character's queue, she is forced into the actors pool and she will not stop acting—and thus the timestep will not terminate—until the action is taken. Whenever the character gets a chance to act, the action manager targets the urgent action.
- *Semiurgent.* If it is a character's turn to act and she has a semiurgent action in her queue, the action manager targets it. However, having a semiurgent action does not force a character to remain in (or re-enter) the actors pool, which means the action may or may not be taken on the current timestep.
- *Nonurgent.* If a queued action is not marked urgent or semiurgent, the action manager may probabilistically target it on a given timestep. As such, it is not guaranteed to be targeted at any point, let alone the current timestep. This probabilistic trigger considers the character's `activity_level` personality trait, which means more active characters tend to dip into their action queues more often. Often, nonurgent queueing occurs for special actions, which can only be taken if they are queued. An example action that is queued nonurgently is `contemplate-serial-mistreatment`, in which a character recalls three past actions by which a specific other character mistreated her. This action is queued on the occurrence of a third such action, but it is

more potent to have it occur some time later, rather than immediately (i.e., urgently or semiurgently). As such, it is queued nonurgently.

- **Priority.** The *priority* of a queued action specifies exactly how urgent it is, according to a numeric score on an arbitrary scale. For example, `escape-burning-building` should have a higher priority then `retort`, otherwise a character in a burning building would attempt to retort before escaping to safety. Priority modeling is commonly used in systems with action queues, especially in game AI (e.g., [232, 1383, 28, 524, 1138]). To be clear, in *Hennepin*, queued actions are prioritized by urgency and then secondarily by priority.

- **Location.** A queued action may be specified such that it can only be taken at a certain location. For example, `construct-frame-for-building` will require the character to be at the lot on which the building is being erected.

- **Date.** Likewise, a queued action may only be viable on some particular future day. For example, if a character plans a birthday party and invites her friends, those who accept the invitation will make a plan (queue an action) to go to the location of the party on the specified date. Of course, it would not make sense to go to the party at a different time, though that could be humorous, and in fact such a humorous effect could be intentionally targeted—for example, if a character is scatterbrained, the action manager could intentionally queue her action on the wrong date, or instead schedule an `arrive-at-party-on-wrong-date` action for the wrong date.

- **Time of day.** Sometimes an action only makes sense at a certain time of day. For example, `set-fire-to-building` and other criminal actions are probably better suited for nighttime. As I explained above, *Hennepin* models day and night timesteps, so these are the two options here.

- **Duration.** Some queued actions may become stale if they are not taken soon enough, and such expiration can be modeled by specifying the *duration* (in terms of the number of timesteps) that a scheduled action should stay in the queue. When a queued action expires, it is removed from the queue.

- **Abandonment conditions.** The *abandonment conditions*, if triggered, cause a queued action to be removed from a character's action queue. Whenever a queued action is about to be targeted, the action manager first evaluates its abandonment conditions, which if triggered will dequeue the action (causing the manager to target a different one instead). These are especially useful for urgent actions that become unavailable upon the storyworld changing in some way, since otherwise the action manager will hopelessly keep targeting the action, even though its preconditions cannot possibly hold.[47] Here are example abandonment conditions for the action `fight-back` (specifying to drop the queued action if the character who instigated the fight is no longer near her opponent):

```
abandonment_conditions=lambda c: (
    not c.instigator.is_at(c.opponent.location)
)
```

[47] When this happens, the action manager instead targets a random generic action for the character and keeps her in the actor pool, but once the pool dwindles down to just her, the simulation reduces to her doing the same generic actions over and over again. I keep this is in there essentially as a visual cue for an authoring error—since the incessant repetition of an action is visually striking in my console output during world generation—but it does produce some amusing scenes. For instance, before I had authored many generic actions, one character could not take his urgent queued action and in fact could only perform `crack-a-joke`, which led to him cracking literally millions of jokes before I manually interrupted the simulation execution. The image of a man nervously cracking millions of jokes is uncanny. In an ironic variant on this scene, a character named Nellie Ancell incessantly performed `extol-virtues-of-value` with regard to the importance of `self-control`.

- **Expiration codes and kill codes.** One authoring pattern is to queue multiple candidate variants of the same kind of response all at once, with the idea that only the appropriate one will actually be selected. For example, a recipient of a `start-fight` action may respond by performing either `fight-back`, `flee-from-fight`, or `talk-it-out`, where the specific appropriate response will depend on the character's personality profile. For instance, these actions could be driven by, and could be used to express, a high `violent` trait, a low `bravery` trait, or a `high` assertiveness trait, respectively.[48] The queueing of this responses can easily be conditionalized, since queuing behavior is defined on the right-hand side of the conditional action effect rules. As such, the personality trait values that I have just mentioned could be referenced in conditions that are attached to the effects that would queue the particular response action that corresponds to that personality profile. In this way, only a violent character would queue `fight-back` and only a character lacking bravery would queue `flee-from-fight`.[49] But what if a character was both violent and not brave? She would queue both of these actions, and because these would likely be urgent, she would probably take both of them. This could be interesting in that it expresses a case of compet-

[48]As I mentioned in Section 11.1.2, one of my major authoring patterns so far is to author actions that are meant to express strong personality traits. Specifically, I scroll through the fifty traits and, for each, think about what kind of actions someone would do if that trait manifested in an extreme way (in either direction). For example, for characters with extremely high `vanity`, I authored an action `admire-reflection-in-mirror`. Such an action is pointless with regard to how it changes the storyworld, but it does a great job of expressing what a character who takes the action is like, which meets Richard Evans's call for actions that vividly express personality [325]. This is where the dramatic charge of the Adams brothers' personality typology is especially appreciated, since their traits are so much more evocative than the Big Five—for example, compare `agreeableness` with `vengeful`. I am also utilizing this authoring pattern for character values, in which the resulting actions are meant to express a strong conviction about some particular issue, such as a `hard work` or `diversity`.

[49]This example works in the style of Richard Evans's modeling of personality traits as conditionals [325], whereby a core manifestation of personality is in character *reactions*, as opposed to actions. In *Hennepin*, both kinds of personality expression are at work.

ing traits with regard to this social context, but in many cases it would not make sense to take multiple response actions that are meant to each be variants on the same archetypal response situation, such as how to respond to a fight being instigated. This is where *expiration codes* and *kill codes* come in handy. When an action is executed, each of its kill codes are retrieved by the action manager, which then proceeds to dequeue any queued action whose expiration code matches the kill code. In the case of the fight response, the action manager could queue each with the same expiration code and same kill code—for instance, attribute the expiration code `99` to each, and also specify a value for their kill codes as `[99]`. Now, when `fight-back`, for instance, happens to be selected by the violent character who is not brave, the kill code `99` will be used to dequeue the queued `flee-from-fight` action. In cases like this, I simply use the location in memory of the object representing the trigger action (in this case, `start-fight`), since that is guaranteed to be unique. I have also used this pattern for more complex coordination, where codes propagate across causal graphs, so that collateral nodes in the graph can cancel one another. For example, a character may attempt to track another character down in order to tell that person something urgent—for instance, that her house is burning down. In this case, two actions are queued at once: `search-for-person` and `report-urgent-situation`, the latter of which requires the two characters to be co-located. When `search-for-person` is executed, it in turn causes the character to queue four more actions: three `go-to` actions by which the character heads to a location where she might find the target character, and `give-up-search`, which is taken after the `go-to` actions are executed, since it has a lower priority than them. Once `give-up-search` is executed, the action manager needs to kill the urgent `report-urgent-situation` queued action, since the target character was not reached (and the urgent action will cause the timestep to never terminate). In order to do this, `give-up-search`'s kill codes must contain `report-urgent-situation`'s expiration code, and this requires the expiration code to be propagated across the causal graph (because `give-up-search` is a niece/nephew of `report-urgent-situation`). Specifically, `search-for-person` must pass on the expiration code to `give-up-search`. While this is a relatively simple case, the same approach could be used to propagate codes across more complex causal structures, and indeed this pattern supports powerful coordination between queued actions.

Action Causality and Causal Bookkeeping

Briefly, I would to summarize how causal bookkeeping works in *Hennepin*. As I explained at length in Section 5.4, causal bookkeeping is a technique by which a simulation (or some secondary system) records causal relations between emergent events as they transpire. Critically, this depends on an explicit and simplified modeling of causality—one that specifically models *contingency* relations—which is atypical of character simulations. Generally, action selection in such a simulation relies on the state of the storyworld, rather than explicit reasoning about past events; while this may produce rich character activity, the emergent causal structure that links past events can become too complex to support narration.

When the state of the storyworld causes every action and is altered as an effect of each of those actions, then a causal structure resembling a dense lattice obtains: every action was caused by nearly every earlier action and caused nearly every later action. In this way, causality for a given action is distributed sparsely across the set of all preceding actions. While this is likely realistic, it is not how causality works in narrative, where narrower and more comprehensible structures obtain instead. As such, for a simulation to be *narratable*—for it to facilitate a curation procedure by which actual emergent narrative may be produced (see Section 4.2.1)—the narrow causal structures of narrative must be identifiable.

I call such structures *emergent contingency structures*, following the *contingency* causal relation identified by narratologist Mark Alan Powell: as opposed to weaker *possibility* and *probability* causal links, contingency obtains when an event clearly causes a later one (in a way that legible to humans). As I also mentioned earlier, the philosopher Abraham Wolf likewise identifies these three links, and additionally a forth one: *chance* [1348, pp. 329–330]. Of course, chance is not a causal event relation, because an event that emerges by chance does not have any explicit relation to earlier events.[50]

In *Hennepin*, action causality may work according to all four of these notions. First, an action may be deemed to emerge by *chance* when the definition of its action type includes no preconditions.[51] Next, an action makes a later one *possible* when its effects change the storyworld such that the latter action's preconditions become satisfied. Third, because of the action manager's satisficing policy, as opposed to an optimal one that would model utility or some other notion of fittingness, the *probability* relation is not really at play in the system, but there are arguable cases. An extension of the authoring pattern of defining a precondition that is simply a probability, such as `random.random() < 0.5`, is to pack in a

[50]Though see my next footnote.

[51]This is not especially precise, since a series of earlier actions would have led to the character who takes the action coming to even exist in the first place, for instance. But is my every action caused (according to a possibility relation) by my birth? And then in turn is my every action caused by my mother's birth? And was that not due to a butterfly flapping its wings, and in turn that butterfly being born, and so on and so forth, etc., and so forth. In a sensitive dynamical system, everything causes everything else, and this produces the kind of dense lattice structure that I have argued is antithetical to narration. Nonetheless, it is fair to argue that nothing can actually occur by chance—that seems to be the case—but I think in this context it is more useful to clumsily conceive of chance actions.

binary condition as well, resulting in examples like this:

```
lambda c: (
    (
        c.insulter.hates(c.insulted) and
        random.random() < 0.7
    ) or
    (
        c.insulter.dislikes(c.insulted) and
        random.random() < 0.3
    )
)
```

All other things being equal, these preconditions provide for a 70% chance of the action being taken if the prospective `insulter` character hates the candidate `insulted` character, as opposed to a 30% chance if the former only dislikes the latter. As such, the earlier action that led to a change in the former's charge toward the latter (such that the threshold for hating was eclipsed) could be said to have made this action more probable.

Finally, the créme de la créme: *contingency*.[52] As I have already explained above, contingency relations in *Hennepin* may be obtain in two ways. First, if an action casts an earlier action in any role—such as a `gossip` action casting an earlier action in its `past_action` role—the later one is treated as having been contingent on the one that it references. Indeed, how could a character have gossiped about some event had that event not occurred? This a stronger relation than possibility—it is an example of contingency. Next, if an executed action effect causes a later action to be queued (regardless of whether the actions' initiators are the same character), the potential for contingency obtains, and the system tracks this by recording that the earlier action is the `trigger` of the queued action. If the queued action is ultimately taken later on, then the system will record the

[52] Again, a contingency relation between events obtains when the former explicitly causes the latter, in a way that can be explained simply.

contingency relation between the trigger and that action. In both cases, *contingent unlocking*—described at length in Section 5.4—is at work.

Contingency relations are stored in the `causes` and `caused` list attributes of concrete actions, which allows the system to traverse both backward and forward across causal links, which in total work to compose directed acyclic graphs (since the `causes` and `caused` lists may each contain multiple actions). This is an *emergent contingency structure*, and it is identified by the procedure of *causal bookkeeping* that I have just explained.[53] Figure 11.3 illustrates an excerpt from an actual emergent contingency structure that obtained in a *Hennepin* storyworld.

As I will explain in the next chapter, this kind of causal bookkeeping makes story sifting a simple matter of retrieval: given some anchor event—what William Labov calls the 'most reportable event' [632], as I explained in Section 4.1.1—retrieve the emergent contingency structure in which it is embedded by chaining forward and backward until causal termination. For example, in the case of the revenge story shown in Figure 11.3, a story sifter might identify `die-in-a-fire` as a most reportable event—for instance, by me specifying actions that are inherently *tellable* [882] according to Roger Schank's notion of *absolute interests* [1103]—and then work backward to retrieve all of the events that caused it. This is an operationalization of William Labov's procedure for *narrative pre-construction*, which I also explained in Section 4.1.1. Alternatively, the story sifter could work back to the primogenitor of the contingency structure and then recursively retrieve all of its descendants, which may result in a larger structure that also contains collateral relations to the most reportable event. In any case, such a structure may serve as material that the sifter can pass on to a narrativizer, which can use it to build a narrative artifact recounting a story centered on the anchor event.

[53]These terms were defined in Section 5.4.

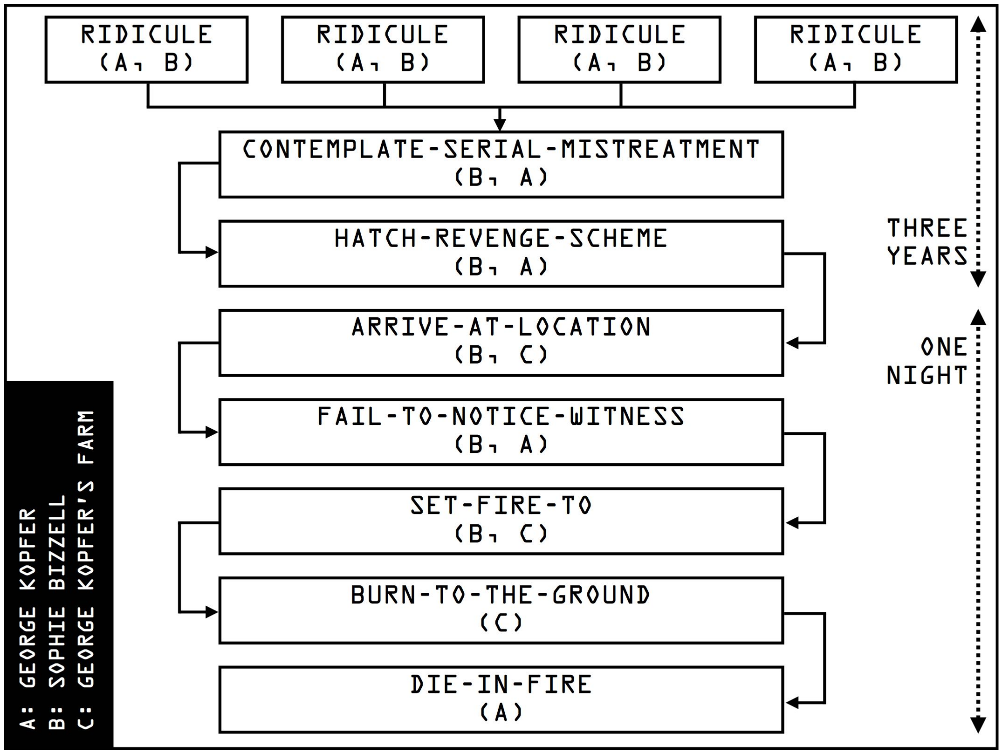


Figure 11.3: An illustration of an excerpt from an emergent contingency structure that was automatically identified in a *Hennepin* storyworld. After multiple years of ridicule by George Kopfer, Sophie Bizzell contemplated the mistreatment one night. Due to having an extremely high value for the `vengeful` personality trait, this contemplation led her to hatch a revenge scheme against Kopfer. A few months later, she decided to enact her scheme and arrived at Kopfer's farmhouse, where she did not notice that he was inside. She set fire the structure, it burned to the ground, and George Kopfer died in the conflagration. Critically, each emergent action was explicitly *contingent* on a previous emergent action (indicated by the solid arrows), which allows for both the emergence of coherent event sequences—what I call *emergent contingency structures*—and the identification of them later on. An automatic story sifter that encounters the `die-in-a-fire` action (which is intriguing in the sense of Roger Schank's notion of *absolute interests*) could retrieve this structure by simply traversing the contingency links that are recorded by the system. In turn, the material illustrated here could be passed off to a narrativizer module that recounts the revenge story—this is how *Sheldon County* works. Finally, what is shown here is only an excerpt, since the causal thread kept going. When Kopfer died, his farmhands were laid off, which led them to look for work, which led to hiring actions, and so forth. In a next authoring phase, I will define actions that correspond to criminal investigation and county trials.

Supporting the Aesthetics of Emergent Narrative

Earlier in this dissertation, I noted that the development of emergent contingency structures over simulated time suggests to me the image of a plantlike growth climbing up out of a backdrop of noise. With regard to *Hennepin*, contingent unlocking drives the plantlike growth and the massive accumulation of commonplace actions—actions caused only according to chance, possibility, or probability—are like the spongy detritus out of which the foliage sprouts. In this way, while contingent actions are the most interesting ones—constituting the skeleton of what may be constructed into an emergent story—commonplace ones provide a backdrop of everyday life that contextualizes the contingent actions.[54] Without this backdrop, the emergent stories resemble nonfiction (see Section 3.1.1) to a lesser degree, and additionally the analogy to worldbuilding (Section 3.1.3) weakens. Moreover, this admixture of everyday occurrence makes the events recounted in an emergent story stand out more—indeed, some characters may never even participate in an interesting event, but instead will carry out a life of monotonous daily existence.

In this sense, the backdrop works as a backbone that props up much of the *aesthetics of emergent narrative* that I defined in Section 3.2: namely, the *aesthetics of the actual*, *aesthetics of a larger context*, *aesthetics of the vast*, and *aesthetics of the improbable*. However, as I have made clear by now, a simulation itself cannot yield these aesthetics, but rather it can enable them at the level of a full-fledged media experience. In my continuing work on *Sheldon County*, I am attempting the capture these potentials, as generated by the underlying *Hennepin*, so that the aesthetics of emergent narrative may characterize the project's procedurally

[54] I plan for a storyworld's history to comprise on the order of a few million commonplace actions, and several thousand intriguing ones. The magnitude of total actions can be (roughly) controlled by two authorial levers: the probability of a timestep being simulated and the probability of a character remaining in the actors pool after taking an action.

generated podcasts. If I am successful in this endeavor, the pleasure of emergent narrative will obtain in that project.

11.2.8 Character Aspirations

Hennepin characters may form *aspirations* specifying what they would like to do in life—for example, a player may aspire to have a child, or found a business, or become a police officer. This system is handled by a module called the *aspiration manager*. Aspirations are defined as a kind of procedural content, and each definition contains the following components:

- **Name.** A human-readable name for the aspiration—for example, `become a police officer` or `get married`.

- **Trigger conditions.** A specification of preconditions that, if triggered, will cause a character to *form* an aspiration. At a technical level, this is defined as a Python lambda expression that takes a character as its only argument. Generally, trigger conditions operate over a character's value system, since beliefs about what is important in life contribute to the determination of what one aspires to do in life. As an example, the aspiration `become a police officer` may be triggered if a given character has a high appraisal of the values `power` and `law`. Unlike action selection, aspiration formation occurs more periodically, on the order of twice a year.

- **Fulfillment conditions.** These conditions define the left-hand side of a fulfillment rule that, if fired, causes the aspiration to be *fulfilled* for the character. This is a Python lambda expression that takes an action as its sole argument. Whenever an action occurs, the aspiration manager uses it to evaluate the fulfillment conditions for every active character aspiration.

Forming and Fulfilling Aspirations

When it is time for the aspiration manager to carry out the procedure for aspiration formation, every character in the storyworld is used to evaluate the trigger conditions for every authored aspiration; in the case that one triggers, that character will form that aspiration. As I have noted, whenever an action occurs, the aspiration manager uses it to evaluate the fulfillment conditions of every held aspiration; if any of these trigger, the corresponding aspirations are fulfilled. It may seem like the manager could simply check whether the *initiator*'s aspirations are fulfilled, but this is not the case, because a character's aspirations may need to be fulfilled by actions that others initiate. For example, if a character aspires to become a police officer, fulfillment will have to come in the form of a `hire` action performed by an employee of the police station (likely the police chief).

Inspiration: *The Sims 2*

Hennepin's modeling of character aspirations is inspired by the *fears and wants* system of *The Sims 2* [793, 146, 1162]. In that game, characters have fears and aspirations that are represented as tree data structures that specify event sequences by which certain fears or aspirations may be realized. For instance, a teen character may aspire to have her first kiss, and so the event sequence would include events like meeting, becoming romantically interested in, and eventually kissing another character. As gameplay proceeds, the system parses the ongoing event stream to check for partial completions of the sequences specified in the aspiration trees. To my understanding, this entails the evaluation of fulfillment triggers that are attached to the nodes of the aspiration trees. If at any point a node is triggered, the system may nudge the simulation toward the event that would lead to the emergence of the next node in that branch of the tree. Once a given sequence culminates, the system triggers content that will showcase a kind of story constituted in the fulfillment of the character's aspiration.

11.2.9 Character Knowledge

Character knowledge in *Hennepin* pertains almost exclusively to *actions*. When an action transpires, all of its participants (including bystanders) form a memory about it, by instantiating a corresponding *knol* whose sole source is the action itself. As I explained in Section 11.1.2, a 'knol' is a unit of knowledge that pertains to a specific individual action.

Knowledge Propagation

In the case that an action relays other actions, knowledge about those actions will be built up as well. If a character has never heard about the relayed action, a new knol is constructed, and the action that has just transpired (the one that relayed the action to which the new knol corresponds) is recorded as the sole initial source of the knol. Alternatively, if a character has already heard about the relayed action, her knol for that action will add the relaying action as a source. If the character had forgotten that relayed action since first learning about it, by this transpiring she will remember it again. In this way, participants in actions form memories about them, and that knowledge may be propagated later on by actions that relay the first one.

While earlier I gave the example of knowledge about an action propagating by a `gossip` action for which the former is a subject, propagation more generally occurs whenever a role with the `action` attribute is cast.[55] Additionally, characters may

[55]It is tempting here to model cases where a relayed action recursively relays other actions. For instance, if someone gossips about someone gossiping, is the action that was gossiped about in the first place relayed to the recipient of the higher-order meta-gossip? I decided that in *Hennepin* it does not, since it seemed like a dangerous rabbit hole to climb into (though one I

recall past events in the course of internal actions, in which case the memory becomes strengthened, since the recollection action will be added to the sources of the knol for the recalled action. Moreover, as I will explain further in Section 11.2.11, artifacts in *Hennepin* may encode knowledge about actions. For instance, if someone writes a note recounting a past action, examining that artifact will cause knowledge of the recounted action to be inferred by the examiner. As such, this is another way that knowledge may propagate.

Character–Location Models

I briefly mentioned earlier that characters build up simple Bayesian models instantiating theories about where other characters tend to spend their time. In such a *character–location model*, every other character in the storyworld is associated with the number of times that she has been located at the various storyworld locales (inasmuch the owner of the model knows that information). Whenever a action is learned about for the first time, the learning character will take every present participant in that action and increment by one the number of times that the character is known to have been at the location at which the action occurred. In this way, someone's model of another character's whereabouts may look something like this: `{Horswill Farm Park: 3, Lessard's Lounge: 8, Law Offices of Mateas and Wardrip-Fruin: 1, Talking Cure: 4}`. As such a model evolves over time, in this computationally efficient manner, a character builds up an increasingly refined sense of where someone may be at any time.

One action that I have authored, called `search-for-person`, relies heavily on this knowledge. In executing this action, the actor heads to the three most likely locations, each in turn, in an attempt to track someone down (usually to impart something urgent, such as the person's house being on fire). Even though this

would have gladly plumbed in my work on *Talk of the Town*).

method is very simple and does not account for a number of things—whether the place still exists, what time of day it is, the last time the person was known to be there—it actually tends to work pretty well. It is intriguing to watch a character successfully track down another by utilizing simple knowledge that was built up in the course of her experience in a storyworld. Knowledge is power.

Memory Salience and Forgetting

Finally, as I noted above, knols have associated *salience* values that capture the strength of a character's recollection of some action. As a new sources for a knol is processed, its salience rules are evaluated, which may trigger updates to the salience associated with a knol. Thus, the more often a character hears or thinks about an action, the more salient it becomes for her. Over time, salience decays as an operationalization of a character's memory of some event fading—if a low salience threshold is crossed, the character will *forget* about the action. The rate of decay depends on the character's `memory` attribute. However, if later on the character hears about it again, she recalls the action and her original sources for it are likewise retrieved.

No Fallibility

Unlike *Talk of the Town*, this system does not model memory fallibility. This design pivot is intended as a simplifying measure, since the fallibility phenomena of my earlier system incurred massive authorial burden (to specify all the different ways each kind of belief should mutate), as I explained in Chapter 9. If I encounter a strong use case for fallibility, I may implement a simpler, more generalized approach—for example, characters could conflate the entities that were cast in the roles of actions they know about, which would be akin the *transference* phenomenon in *Talk of the Town* (see Section 9.2.12).

11.2.10 Character Internal Worlds

I have already noted several times that I am interested in modeling the internal lives of characters. As I discussed in Section 7.3.15, in *World* this notion was briefly explored through the simulation of characters grieving, regretting, wondering, worrying, and reminiscing.[56] In *Hennepin*, inner life transpires as characters take the various internal actions that I have authored—so far, these are:

```
admire-the-sunrise
admire-the-sunset
agonize-over-hair
agonize-over-outfit
agonize-over-quality-of-work
consider-a-new-perspective
contemplate-prolonged-unemployment
contemplate-serial-mistreatment
crack-under-pressure-at-school
crack-under-pressure-at-work
cringe-over-past-action
cry-over-past-action
decide-to-confront-person-over-past-action
envy-more-popular-kid
envy-more-successful-sibling
envy-spouse-of-love-interest
fantasize-about-life-with-love-interest
gaze-upon-the-stars
obsess-over-love-interest
stare-at-self-in-mirror
vent-about-mistreatment
```

[56]While *Talk of the Town* essentially does not simulate character internal worlds, with Tyler Brothers I developed an extension to it by which characters would generate streams of consciousness in response to emotionally charged environmental signals. The farthest we got in this direction was an initial videogame prototype, called *Juke Joint* [1042], about a haunted jukebox that changes people's lives (according to the particular lyrics that it emits).

Related Work

Briefly, I would like to outline some earlier work on this topic. At a high level, inner life pertains generally to the autonomous processing of ongoing subjective experience. A multitude of earlier projects have explored agent real-time perception (often visual processing) in real or simulated environments [503, 1109, 334, 1327], including perception systems for NPCs in combat and stealth games [677, 265, 904, 529]. More relatedly, projects on autonomous characters—from *Tale-Spin* [823] to the Oz Project [86] to *Façade* [775]—have modeled environmental perception for expressive purposes. While these projects aim to generate optimal or believable agent behavior in particular environmental contexts, in *Hennepin* I am more interested in how a character's subjective inner world may buzz as a largely involuntary dynamic of how the world is and who the character is (namely her personality profile).

Given this difference, let us turn then to the expressive modeling of agent subjectivity. While in Section **??** I reviewed the literature on modeling subjective beliefs, here I would like to consider systems that have modeled memory, thought, and other internal phenomena. Interesting early systems in this vein include: the curiously forgotten *Aldous* [695, 696, 697, 698], an autonomous agent that processes its environment to generate emotionally charged responses; *Erma* [192, 193], an obscure early chatbot that deeply models the subconscious layers of processing that mediate speech (to introduce disfluencies, for instance); *Parry* [198, 608], Kenneth Colby's famous model of clinical paranoia; and *Cyrus*, which models agent reconstructive memory in a simulated politician [613]. *Daydreamer* is a later system that generates daydreams by reflecting on past experiences encoded in a knowledgebase [863, 862]. More recently still, *Griot* uses conceptual blending to generate poetry that proceeds from evocative user input [451]. Follow-up work extended the system to produce daydream-like output that generates character thoughts, outlined in the third person, that are also guided by user input [1377]. *V-Daydreamer* has similar aims to these systems, but produces visual output [938].

Generally, *Hennepin* departs from these projects by more deeply *situating* its characters—internal phenomena is not the entire simulation, but rather one thing that can happen to characters as they go about their tiny little lives.

11.2.11 Artifact Phenomena

As I have already explained, certain actions cause new *artifacts* to be created in the storyworld. For example, `write-a-poem` produces a poem and `file-deed-on-parcel` yields a deed. An artifact has the following attributes:

- **Description.** A human-readable *description* of an artifact, for example, `a child's drawing of [write-a-poem] by Amanda Muskopf`.[57]

- **Location.** Artifacts are always positioned at specific *locations* (either in a building or on a lot). An artifact may only be examined by characters who are at its same location.

- **Durability.** An action may *destroy* an artifact. For instance, if a character performs `burn-artifact` with a poem cast in the `artifact` role, then that poem would likely be destroyed according to an executed action effect. Specifically, such an effect would specify the chance of an artifact being destroyed according to its *durability*, which is defined by an attribute that takes a numeric value. For example, a poem is less durable than a ring. If a calamity occurs at a location, *all* of its artifacts may be destroyed—if a house were to burn down, as in Figure 11.3, all the artifacts inside the house could

[57] This was a drawing by a young girl that depicted her mother writing a poem, an action she had heard about from her brother.

potentially be destroyed. Finally, an artifact weathers over time, which is modeled by its `durability` numeric value gradually lowering.

- **Detectability.** When a character is located near an artifact, she may take the action of examining it, but only if she detects it; the *detectability* of an artifact is another attribute that is represented by a numeric value. This is handled by a simple probabilistic precondition on a catch-all action `examine-artifact`, which gates examination by using the detectability value to generate a detection chance. Action effects may also modulate the detectability an artifact—for instance, if a character buries an artifact on a lot, it becomes far less detectable.

- **Intrigue.** Some artifacts are more interesting than others, and this notion is captured by an artifact's *intrigue*, which is also represented as a numeric value. For instance, a bloody knife is generally more intriguing than a paycheck. As with detectability, intrigue is referenced in a probabilistic precondition on the `examine-artifact` action.

- **Origin.** The *origin* of an artifact is the action by which an artifact was created—for instance, `write-a-poem` in the case of a poem.

- **Destruction.** The *destruction* of an artifact is the action by which an artifact was destroyed—for example, `burn-artifact` in the case of an incinerated poem.

- **Moves.** Certain actions may cause an artifact to be *moved* to a new location. For example, `move-belongings` causes all of the artifacts located in one house to be moved to another house. Whenever an action moves an artifact, it is added to that artifact's `moves` list attribute.

- **Examinations.** Whenever a character carries out an *examination* of an artifact, the corresponding `examine-artifact` action is added to the artifact's `examinations` list attribute.

- **Inscriptions.** As I explained in Section 11.2.9, an artifact may encode knowledge about past events, which a character learns about upon examining the artifact. Typically, an artifact encodes knowledge about its origin—for example, information capturing a `file-deed` action will be encoded in the deed produced thereby. Additionally, certain actions may cause information about additional past actions to be *inscribed* into an artifact after its time of origination. For instance, the action `journal-about-event` causes information about a subject action to be inscribed into a journal artifact that already exists. Whenever an action inscribes new information into an artifact, it is added to the artifact's `inscriptions` list attribute.

- **Transmissions.** An artifact's *transmissions* are all of the actions about which knowledge will be transmitted upon a character examining the artifact; these are stored in a `transmissions` list attribute.[58] Since there is currently no modeling of partial information, a character who learns about an action has full access to all of the pertinent information, including all of the entities bound to all the action's roles and its location and time of occurrence. If I encounter a use case for partial information, I will likely introduce a procedure for determining exactly what information about an inscribed action is encoded in an artifact (and thereby what information will be transmitted upon examination). In Section 9.3, I will outline some emergent phenomena that demonstrates the narrative potential of encoding

[58]Note: transmissions are a superset of all inscribed actions and all the other actions that are recounted by an artifact.

information about the past in artifacts.

- **Effects.** Finally, when an artifact is examined by a character, *effects* may be conditionally executed. Technically, these effects are attached to the `examine-artifact` action definition (with preconditions on each specifying an artifact type), but for clarity I describe it here. In this way, examining an artifact may cause an action to be added to a character's action queue. In the next section, I describe how examining a treasure map queues a follow-up action by which the map examiner may dig up the treasure at the location specified on the map. Of course, these conditional effects on artifact examination may specify other concerns such that not every character will react the same way to examining an artifact. For instance, if a character were to read her partner's diary and learn of an extramarital love interest, she might queue an action to question her spouse. However, the subject of that love interest, upon reading the diary, would react differently, perhaps by increasing her spark for the diarist. Such variation would be specified in the condition logic of the various alternative effects, following the same patterns that I illustrated in the various code listings included above.

11.3 Emergent Phenomena

Just as I did with *World* and *Talk of the Town*, I will now report some examples of interesting emergent phenomena that I have encountered in *Hennepin* storyworlds. Whereas *World* primarily generates emergent images, as discussed in Section 7.4, and *Talk of the Town* only produces *emergent scenarios*, as explained in Section 9.3, *Hennepin*'s storyworlds reliably yield *emergent storylines*. To be precise, what emerges are contingency structures, not fully formed storylines—as

I have established repeatedly by evoking the ideas of Hayden White, Arthur Dantos, Epsen Aarseth, and others, actual storylines only obtain through an act of narrative construction. In the case of emergent narrative, such an act will more specifically be a procedure of curation. Thus, in this section I am presenting cases of emergent contingency structures that have enough causal coherence to greatly facilitate the work of a story sifter and narrativizer. For convenience, I will use terms like 'emergent storyline' to refer to these structures, but really I mean 'emergent contingency structure'.

Human Dramatic Situations

When I began authoring actions for *Hennepin*, I decided to target a particular pocket of narrative possibility space: *revenge stories*. More specifically, I authored toward the emergence of revenge by arson, which the excerpt sequence of Figure 11.3 exemplifies. At this time, I was rearing from the trial of attempting to do automatic curation in *Talk of the Town*, a problem of which is that the events that are modeled in that system are generally not inherently interesting. It is hard to tell a good story about a business's hiring process, for instance, and that is the kind of emergent material that the engine tends to produce. As such, I figured that I would start my work on *Hennepin* by targeting emergent material that is unequivocally dramatic, and revenge by arson is certainly that.[59] Once I had authored all the component elements that would compose a storyline concerning revenge by arson, I booted up the simulation engine to test whether the scenario could actually emerge given what I had authored. Remarkably, even though I had authored only a few action definitions, a storyline emerged that surprised me.

[59] In hindsight, I was intuitively veering toward Roger Schank's absolute interests, a notion that I have outlined several times now. In the case of revenge by arson, Robert Wilensky's variant phrase "human dramatic situation" [1340, 1341] is a particularly apt descriptor.

The Arsonists

Among this particular county's settlers were a husband and wife named Julius and Marie Eckert, each of whom had high values for the `cruel` personality trait. Julius owned a farm and employed a farmhand named Roy Champ, who was a frequent target of Julius's `ridicule` actions during workdays in the field. At this time, there were only a few authored actions for characters to choose from, and so Julius spent each workday incessantly ridiculing the farmhand Roy Champ. Meanwhile, Julius's wife Marie spent most of her days calling on a neighbor named Catherine Lumpp, who was the wife of another local farmer. Like her husband, Marie had a predilection for bullying others, in her case the target being the unfortunate Catherine Lumpp.

Eventually, both Champ and Lumpp would come to perform the `contemplate-serial-mistreatment` action. Remarkably, both characters had high enough `vengeful` trait values for such contemplation to cause a `hatch-revenge-scheme` action to be added to their respective queues. While the schemes were not hatched on the same timestep, improbably they would be *enacted* on the very same night. This caused both characters to arrive at the Eckert farmhouse, which was empty, as Julius and Marie had gone to a tavern in the county. Stunningly, neither Champ nor Lumpp noticed the other being there—this was modeled by each performing the action `fail-to-notice-witness`. Next, each took the action `set-fire-to`, which queues a series of actions in turn: `burn-to-the-ground` (with a null initiator) and `escape-burning-building` for each character that is in the building besides the arsonist. As such, the `escape-burning-building` action was queued for both Lumpp and Champ.

Critically, `burn-to-the-ground` was queued twice, once for each `set-fire-to` action, and in each case the non-arsonist was pre-bound to a role called `potential-`

victim. When burn-to-the-ground occurs, its effects queue die-in-a-fire for all characters bound to potential-victim, though this is conditional on those characters still being located inside the building. Additionally, using kill codes, burn-to-the-ground cancels any queued escape-burning-building actions. As such, with both escape-burning-building and burn-to-the-ground (and thereby the latent die-in-a-fire) queued, survival becomes a simple matter of which outcome the action manager randomly selects for targeting first: if an escape-burning-building action is targeted, its initiator escapes the building; if burn-to-the-ground is randomly selected instead, then it is executed, which queues die-in-a-fire, which kills the initiator. In this case, both of the burn-to-the-ground actions were targeted first, and thereby neither Champ nor Lumpp were able to escape, and so both died in the conflagration.

While I authored to enable the emergence of storylines concerning revenge by arson, this I had not anticipated. The result was a simple reduplication of the same exact contingency structure—each example was composed of the same action types, though retargeted to different characters and storyworld contexts—but the improbable lockstep interaction produced something striking (and surprising). This tale of two arsonists was a story of karmic coincidence, with the culprits each receiving a taste of their own deadly medicine at they administered it. The takeaway for me was that the potential for interaction between emergent contingency structures produces a combinatorial explosion of the narrative possibility space, introducing fundamentally new kinds of emergent storylines that even I cannot anticipate. This is what emergent narrative is all about, and this initial test case was very exciting for me.

Storyline Interlocking

I now realize that this structural interaction works by the same principal as scenario stacking in *Talk of the Town*, which I described at length in Section 9.3 and in the opening of this chapter. An example of scenario stacking is a *family rivalry* stacking atop a *business rivalry*, and in turn this combination stacking onto a case of *love* between two characters in those families. The resulting composite scenario might be called *forbidden love*, but I had no intention of authoring toward the emergence of such a scenario—it was itself a dynamical emergent byproduct of the mechanical interaction of the specific kinds of scenarios that I *had* intentionally targeted (family rivalry, business rivalry, love).

While *Talk of the Town* generates scenarios, *Hennepin* produces action sequences, and so this kind of combination in this engine concerns the interaction of concrete structures, not just scenarios. I am very excited about the potential for wild interlocked emergent action sequences that combine multiple discrete storylines—something like the procedurally generated equivalent of a Thomas Pynchon storyworld. Let us call the phenomenon *storyline interlocking*.

The Picnic

In another case of storyline interlocking, a character John Lenning's disparaging of `family` values at a park led two other men to independently (and ironically) pick a fight with him. As John performed `fight-back` in response to both instigations, two fights broke out simultaneously at Milot Park, each pitting Lenning against a family-oriented foe, with the blows being interleaved. A bystander, Susan Ask, watched both skirmishes, presumably in horror.[60]

[60] Another humorous scene centered on value systems concerned one Daryle Guffey, who showed up at the Main Street Brewery in the town of Patience apparently with the sole purpose of denouncing the value `merriment`. This was entirely coincidental, since the action manager had no means with which to associate merriment and breweries, but apophenia takes over.

The Treasure

Hennepin's rich modeling of artifact phenomena also generates some interesting narrative potentials. One evocative pattern concerns artifacts that encode information about past events that no living character knows about. As a simple proof-of-concept test of this idea, I defined a series of actions that led to the following emergent sequence. In the 1850s, at the beginning of a county's history, a character named William Weismantel performed an action `bury-treasure`, whereby a treasure artifact was buried at a secluded lot in the county.[61] As an effect of this action, the treasure artifact's `detectability` attribute value was set to `0.0`, making the artifact totally undetectable, even to characters positioned on that lot. As one of the `bury-treasure` action's effects, a prospective follow-up action was added to Weismantel's action queue: `draw-treasure-map`. When Weismantel performed this action in turn, a map artifact was created as a result. Critically, this map artifact included the `bury-treasure` action in its transmissions, which meant that examining the map would transmit knowledge about where the treasure was buried. As an effect of `draw-treasure-map`, a `hide-artifact` action was in turn queued for Weismantel, with the treasure map pre-bound to an `artifact` role and Wisemantel's house pre-bound to `hiding-place`—for this proof of concept, I hardcoded that one always hides an artifact at one's home. As an effect of `hide-artifact`, the treasure map's detectability was set to a very low value, but critically one that was nonzero; its intrigue value was set to be extremely high.

From this point, the decades passed by and eventually Wisemantel died. Because he never told anybody about his treasure map, and because nobody ever detected it in his house, upon his death no living character in the county had

[61] In my hasty crafting of this proof of concept, this action actually causes the artifact to exist. Of course, it would make more sense for the treasure to originate in a natural way, in which case its acquisition would enable the burying action.

knowledge of his past actions pertaining to the treasure, and thus nobody knew about the treasure. But while no character had knowledge of these events, information about them was still inscribed in the storyworld through the hidden treasure map, which encoded the `bury-treasure` action.

A few years after Wisemantel's death, a character named James Bellard moved into his former home, and some time after that he happened to discover the treasure map, which was still hidden in the house. At a technical level, this occurred when the action manager randomly targeted `examine-artifact` with the treasure map randomly bound to its `artifact` role (because the candidate pool includes all artifacts at the prospective initiator's location). Critically, the preconditions held, which was remarkable because the map's `detectability` numeric value was very low, making detection unlikely. Though the system does not record this information, it is very likely that the action manager had unsuccessfully targeted this action and binding many times prior, only to have the preconditions fail as a modeling of the artifact not being detectable at that time.

When Bellard examined the artifact, an effect caused a new action to be added into his queue and marked urgent: `dig-up-treasure`. Later in the timestep, the action manager targeted this action, causing Bellard to move to the (still vacant and secluded) lot in the county where the treasure was buried. Upon performing `dig-up-treasure`, an effect of the action modified the detectability of the treasure artifact to a very high numeric value representing maximum detectability—this models the unearthing of a buried artifact. Additionally, another effect of `dig-up-treasure` urgently queued `hide-artifact` for Bellard, with the treasure bound to the `artifact` role and Bellard's house bound to `hiding-place`. Finally, by performing this action, Bellard hid the treasure in his home, where its detectability was set to a low level.

Artifact-Driven Storylines

This is where this story ends, since it was only a proof of concept. Nonetheless, it makes for an intriguing little emergent tale, and more importantly it suggests the feasibility of a larger narrative possibility space concerning sought-after artifacts that are hidden and hunted. My favorite aspect of this tale is that knowledge of the treasure did not exist in any living character's mind for a number of years (between Wisemantel's death and Bellard's discovery of the map). What if Bellard's house had burned down, destroying the map? In that case, the treasure would have lied buried, with no record of it existing, except for the undetectable treasure artifact itself. This gave me the idea of having routine actions potentially causing buried artifacts to be unearthed. To model this, I added an effect to `lay-foundation-for-building` that causes any artifacts located on the lot in which the foundation is being laid to have their `detectability` values raised, thereby 'unearthing' them. Thus, *Hennepin*'s narrative possibility space now contains an emergent storyline in which a treasure map is destroyed, but years later the forgotten buried treasure is serendipitously unearthed as part of the routine work of building a house. I have tried a couple different seeds, but so far I have not encountered an emergent sequence of this sort. This could also be implemented as an effect of county work crews preparing lots to serve as road segments, which would cause the construction of roadways to potentially unearth buried artifacts.

More broadly, the fact that examining an artifact may trigger arbitrary effects means that this kind of sequence has narrative potential far exceeding mere tales of treasure. For instance, I can imagine the secret artifacts anchoring stories of blackmail, or Cohen brothers-style escapades. To return to the notion of emergent interaction between emergent storylines, there could be a case of evidence of arson being kept somewhere hidden, and thereby a story about hunting down a secret artifact could interlock with an arson tale like the one recounted above.

Feedback Mechanisms

One of my major design goals for this engine is to implement feedback mechanisms that cause earlier emergent events to alter the course of a storyworld in potentially dramatic ways. This is in line with my ideas on simulation feedback that were presented in Section 4.1.6. Specifically, in *Hennepin*, this would mean that characters may take actions that drastically change how the county looks, or even works, thereby pushing a particular storyworld into its own sector of the simulation engine's possibility space.

A concrete example of this pertains to characters starting new towns in a county. As I have explained in this chapter, characters form value systems about what they believe is important in life. If a group of characters discover that they each hold the same extreme beliefs about a particular set of values, they may decide to form what they view as a utopian town—one built on the ideals they hold in common. As an example, let us say that a group of characters all believe in `law` and despise `merriment`, so they start a town rooted in these principles; let us call it Temperance. As I have mentioned above, *Hennepin* characters hold town hall meetings in which they may propose and vote on new legislation—if the legislation passes, it can actually affect how life in that town is simulated. For example, the residents of Temperance may decide to outlaw drinking in the town, which would be modeled as a new prohibition statute whereby character drinking actions are now illegal. Now, when a police officer learns about a drinking action in the town, through the standard mechanisms for knowledge propagation, an effect conditionalized on there being a prohibition statute causes the officer to queue an `arrest` action accordingly, which means the drinker will be arrested if she ends up at the same place as the officer (or alternatively the officer could work to track down the drinker). Thus, through a feedback mechanism whereby characters actually change how the simulation works, individual storyworlds may propel into novel (and potentially strange) pockets of the possibility space. To be clear, this particular example has not been implemented yet, but it would be possible to author given how the statutes system works.

Emergent Images

Lastly, I will conclude in this section with a few emergent images, in the style of the evocative situations that I recounted in Section 7.4 with regard to *World*. The first concerns one Edward Runyan, mayor of a town called Runyan that he founded early in a county's history. Runyan was a hard worker and a hard drinker who spent most of his days at the town hall performing `work-hard-at-work` and most of his nights at home performing `have-a-drink`. Occasionally, he would get into arguments with Rose Vaillancourt, his secretery, on the merits of `stoicism`, which Runyan valued highly. In one evocative nighttime visit to the Vaillancourt residence, the stoic Edward Runyan performed only these three actions:[62]

```
ID: 8529
Name: have-a-drink
Date: Night of October 1, 1853
Location: Vaillancourt residence (158 Runyan Avenue)
Drinker: Edward Runyan

ID: 8532
Name: examine-artifact
Date: Night of October 1, 1853
Location: Vaillancourt residence (158 Runyan Avenue)
Examiner: Edward Runyan
Artifact: a child's drawing of Rose Vaillancourt
```

[62] This is an excerpt from the console logging that is displayed in the terminal window during world generation.

```
ID: 8533
Name: discern-knowledge-transmitted-by-artifact
Date: Night of October 1, 1853
Location: Vaillancourt residence (158 Runyan Avenue)
Artifact: a child's drawing of Rose Vaillancourt
Learner: Edward Runyan
Past Action:
        ID: 7734
        Name: draw-a-picture-of
        Date: Day of February 27, 1853
        Location: Vaillancourt residence (158 Runyan
                        Avenue)
        Picture: a child's drawing of Rose
                     Vaillancourt
        Drawer: Agnes Vaillancourt
        Subject: Rose Vaillancourt
```

Another character, the first to arrive in his county, happened to cap off the completion of a difficult project with a celebratory drink (a notion not modeled in the action precondition logic, though it works nicely by an apophenia hack):[63]

```
                        ** Day of May 22, 1839 **

17: [lay-foundation-of-house] by John Lanclos at a
      spot on public land

                        ** Day of May 23, 1839 **

18: [build-frame-for-house] by John Lanclos about
      Unoccupied building at a spot on public land

                        ** Day of May 24, 1839 **

19: [raise-roof-on-house] by John Lanclos about
      Unoccupied building at a spot on public land

20: [have-a-drink-during-the-day] by John Lanclos at a
      spot on public land
```

[63] This was an earlier version of the worldgen printout.

Finally, the tragic case of one John Putterman. Putterman had high trait values for both `emotionally_obsessive` and `love_propensity`, which unlocks the internal action `obsess-over-love-interest`. In high school, John Putterman fell in love with a classmate named Sallie Koppinger, but nothing ever materialized between the two and Koppinger ended up leaving town when they were in their twenties. All throughout his life, Putterman constantly performed `obsess-over-love-interest` with Koppinger as its absent subject. Moreover, this recurring action was actually part of a tragic feedback loop: when a character obsesses over a love interest, the spark for that person increases, thereby reinforcing the obsession further. In this way, while his spark for her did decrease over time, the decay rate was no much for the rapid spark increases caused by his constant obsessing. Finally, on his death bed, Putterman, now almost fifty years removed from Koppinger's departure, performed one final action:

```
>>> print john
John Putterman, 1840-1909
>>> print john.death
Death of John Putterman in 1909
>>> print john.love_interest
Sallie Koppinger, left town in 1865
>>> print john.actions.taken[-1].outline
ID: 72735
Name: obsess-over-love-interest
Date: Day of January 30, 1909
Location: Putterman residence (705 6th Street E)
Thinker: John Putterman
Love Interest: Sallie Koppinger
```

Chapter 12

Looking Ahead: Sheldon County

The goals we set for the system are heroic, and demand heroic test cases.

Sheldon Klein

In the fall of 2016, Amazon announced the Alexa Prize [973], a competition that called for university research labs to develop AI technology for the Alexa conversational platform. Amazon was to admit ten entrant teams, each of which would receive a $100,000 grant, Alexa-enabled devices, and free access to Amazon Web Services, a cloud-computing platform. At the time, my coadvisors, Michael Mateas and Noah Wardrip-Fruin, and I were between two rejected grant proposals that each pitched work related to a planned extension to *Talk of the Town.* Michael heard about the Alexa Prize from Ashwin Ram, his former George Tech colleague who was running the competition at Amazon, and he proposed a meeting with Noah and me to brainstorm about a possible entry.[1] I was working as a teaching

[1] Ashwin Ram is another one of Roger Schank's students, like Jim Meehan and Natalie Dehn and others who have appeared at various points in this dissertation, including my Michael's coadvisor Jaime Carbonell. Ram's own dissertation project was a story understanding system called *Aqua* [971]. It appeared toward the end of the Yale AI Project, just as Schank was heading to Northwestern University to found the Institute for the Learning Sciences.

assistant, and so a funded project sounded quite appealing.

In Ram's call for entries, he explained that the specific task of the competition was to develop a speech-enabled bot capable of conversing on "popular topics" for an extended period of time:

> Today, we are pleased to announce the Alexa Prize, a $2.5 million university competition to advance conversational AI through voice. [...] The challenge is to create a socialbot, an Alexa skill that converses coherently and engagingly with humans on popular topics for 20 minutes. We challenge teams to invent an Alexa socialbot smart enough to engage in a fun, high quality conversation on popular topics for 20 minutes. [...] Participating teams will advance several areas of conversational AI including knowledge acquisition, natural language understanding, natural language generation, context modeling, commonsense reasoning and dialog planning. Alexa users will experience truly novel, engaging conversational interactions. [970, n.p.]

Adam Summerville and I had recently explored a technical approach to natural language understanding that made use of my authoring tool *Expressionist* [1220, 1050, 1049], which I had already been using to drive natural language generation in a testbed extension to *Talk of the Town* [1045]. In turn, I had developed a dialogue manager that could integrate both systems [1059], and so we felt that we had a good technological substrate in place for a proposal. A key feature of this technology was its *authorability*—the framework was designed to allow human authors to inscribe artistic intent into (and exert authorial control over) a generative system for procedural conversation. As such, while we figured most entrants would propose less authorial systems—likely ones that would automatically mine knowledge about popular topics to then render that knowledge in natural language—our spin would be to leverage our authorial technology to enable compelling interactions driven by artistic and aesthetic considerations.

At the Expressive Intelligence Studio, we develop novel AI technologies, but we do so in the service of *media works*—this is why 'studio' is in the name of our group, and it is why our department is called 'computational media'. Having this predilection (and mission), our first step was to frame the Alexa Prize call in terms of a prospective work of media, which led us to ask ourselves: what kind of compelling media experience would center on conversation on popular topics? The more we thought about it, however, the less sense the whole competition made to us. What are these 'popular topics' and why would anyone want to discuss them with a computer? The media experience that this framing suggested seemed to us to be either harrowing feats of endurance—*Talk About The Kardashians With A Computer For Twenty Minutes Straight*—or, at best, redundant undertakings. One who wishes to talk about the Kardashians will have little problem doing so, since there are plenty of humans who gladly provide the service. Moreover, humans are almost certainly better than computers at talking about the Kardashians. What is the point of computationalizing a banal everyday experience that has already been mastered by humans? Of course, it could be interesting to subvert that experience or to computationalize it with a peculiar procedural aesthetics,[2] but this was clearly not what Amazon was after and we struggled to excite ourselves about the idea of procedural conversation on popular topics.[3]

Instead of abandoning the prospect, we decided to brainstorm about how we could build an actual media experience that would be novel, compelling, and uniquely suited to the Alexa platform. How would one work? First, the experience would be strictly audio-based, with no screens. In consideration of this, our conversation shifted to a discussion about how we each engage with audio already. Someone mentioned the experience of listening to the radio in the car, and the specific pleasure of serendipitously encountering a radio play on a road trip. I

[2] I can for instance imagine experimental game designer Pippin Barr answering the Alexa Prize call in an unsettling way.

[3] To be fair, this is a decidedly mediacentric appraisal of a competition with primarily technological aims. More on this in footnote 6.

explained that I enjoy podcasts but only listen to them in two recurring contexts: doing dishes and working out. In each of these contexts, I am not free to engage with screen-based media, because I cannot maintain eye contact due to either looking at what I am doing with my hands (dishes) or turbulent body movement (I do a hill workout). Moreover, while doing dishes, my hands are not free to control any kind of input device, but I could use my voice. How could we design an experience for dishwashers?

Somewhere in the discussion, Michael had a brilliant idea for a computational audio experience that would leverage *Talk of the Town*: a procedural episodic radio play in the style of *A Prairie Home Companion* (1974–2016) [647, 646]. Rather than constructing stories about an invented town of fictional characters, a system could recount stories that have actually emerged in a *Talk of the Town* town. Moreover, every listener could have her own personal storyworld, each being driven by a distinct simulation instance, which would suggest an ontology in which the town lives on her Alexa device. It would be like an ant colony, but with orders of magnitude more intrigue.

How would a system know what stories to tell? As Michael noted, one prospect would be to automatize the kind of narrative excavation that I carry out in my capacity as *Bad News* wizard—this would likely result in the technical approach that I demonstrated at the beginning of the previous chapter. In fact, this was the task of 'story recognition' that we had already identified in our paper from the previous year, "Open Design Challenges for Interactive Emergent Narrative" [1058].[4] Once the material was excavated, a natural language account could be generated

[4]As I explained in Section 5.5, under the influence of Hayden White, starting with this dissertation I have decided to use the term 'story sifting' instead of 'story recognition'. The reason for this is that stories do not exist fully intact in (storyworld) chronicles, including simulation traces. To assume otherwise is to commit the crime of 'narrativism' (Espen Aarseth's term), as I explained at length in Section 4.2.1.

using the technology that we had already developed, namely *Expressionist*. What we were now brainstorming in this meeting was an archetypal work of *curationist emergent narrative*: an automatic *story sifter* would excavate promising material from the accumulated history of a simulated storyworld, and a *narrativizer* would then construct a narrative artifacts out of that material, which would themselves be mounted into a media experience taking the form of a radio play. Our working title was *Procedural Wobegon*.[5]

This idea was exciting, but as initially formulated, it would not feature the procedural conversation that was at the heart of the Alexa Prize call (and that is the paradigmatic Alexa interaction). Of course, the way to leverage this interactivity would be to make the radio play interactive. Maybe the listener could interrupt the narrator to ask questions—for instance, an inquiry as to the story of a background character in a scene. Alternatively, or additionally, there could be choice points within or between episodes about what or whom the listener would like to hear about next. In this way, the resulting narrational style would likely become a hybrid between that of a radio play and of conversational storytelling, perhaps with some elements of *branching narrative* [1080, 856].

We were excited by our new idea, but it had obviously deviated from the spirit of competition. What we had conjured was not centered on conversation around popular topics—it in no way pertained to the Kardashians—but it seemed like a strong vision for an experience that would truly leverage the Alexa technology. Michael and Noah figured that Amazon might be interested in supporting the idea to some extent, perhaps by providing hardware and a development kit, even if it was too far off to be accepted as an entrant in the actual Alexa Prize competition. Michael wrote to Ashwin Ram directly to pitch *Procedural Wobegon*, but Ram

[5] *A Prairie Home Companion*'s storyworld is a fictional Minnesotan small town called Lake Wobegon, which is also the setting of Garrison Keillor's spin-off novel *Lake Wobegon Days* [574].

expressed that it was definitely too removed from the call to be an entrant, and moreover Amazon would not be able to support the project in any other way.[6] The idea faded and eventually we attempted to submit a revised form of an earlier National Science Foundation grant proposal, but as I mentioned at the beginning of this chapter, that would ultimately be rejected. Luckily, Noah found another funding opportunity that would cover the remainder of my PhD tenure, but it was one that was adjacent to my research.

A year or so later, I began exploring the integration of fine-grained character actions into *Talk of the Town*, a journey I explained in the previous chapter. This was the first stage of my original dissertation project, which concerned the computational modeling of *autobiographic storytelling* and *persuasive storytelling*—this meant generating the kinds of life stories that people construct to make sense of themselves, and also stories that people tell as a way to change the world (particularly to persuade others). My plan was to have simulated characters accumulate knowledge about what has happened in a storyworld and then curate that knowledge to tell stories about what has happened. In procedurally generating either an autobiographic story or a persuasive one, the storyteller would pursue a goal by targeting a particular *recipient response*—such as someone deciding not to trust someone else—and then chain backward from that response to sift out relevant story material and narrativize it in support of the storytelling goal.

While I was highly intrigued by the notion of modeling this storytelling phe-

[6]Incidentally, UC Santa Cruz would produce an Alexa Prize entrant in the form of the *SlugBot* team [136], led by student Kevin Bowden and composed also of other members of Marilyn Walker's Natural Language and Dialogue Systems lab. To be clear, I do think that this work is valuable and that the Alexa Prize is a good initiative for conversational AI. My qualm is with the notion that procedural conversation around popular topics is a compelling utilization of the Alexa platform from the standpoint of producing actual media experiences. As such, I believe that this work is advancing the cutting edge of (task-driven) conversational AI, but not computational media, and the usage of terms like 'engaging' and 'experience' (in the call [970] and in a recent competition report [973]) suggest an unexamined conflation of these concerns.

nomena, which has not been done before in the history of story generation, I did not have any ideas about a media experience that would be enabled thereby. I could imagine *Procedural Wobegon* characters telling stories to one another, but what I was aiming at in this project was an extremely deep dive into the phenomenon, which would take me far beyond the modeling that would be necessary to support the kind of diegetic storytelling (a type of *embedded narrative* [1063]) that one encounters in media. One prospect was to develop a videogame based on this phenomenon—a fully computational *Bad News*-like, perhaps—but that seemed out of scope given my targeted graduation date. After some soul searching, in which I almost changed my topic to an historical endeavor, I decided that I wanted my final UC Santa Cruz project to be mediacentric, meaning I would proceed from a vision for a media experience, rather than from a technology that could conceivably drive a nebulously defined media experience.[7] This move oc-

[7] If I pivoted to the historical project, my dissertation was going to be titled *Inside Computational Media: Five Histories Plus Reimplementations*. This would riff on Roger Schank and Chris Riesbeck's 1981 classic *Inside Computer Understanding: Five Programs Plus Miniatures* [1107], which its authors later described in this way: "That book contained two kinds of material. First, there were substantial chapters summarizing doctoral work on various models [...] developed at Yale. These chapters were written by the original researchers and described both the theoretical aspects of their work and the practice details. Second, each chapter was accompanied by miniature or 'micro' versions of each AI program discussed. These Lisp programs ranged from 5 to 15 pages long, including substantial commentary [...] The intent was that the programs would capture the bare essentials of the doctoral work, while still be short enough and simple enough to be easily understood and modified by anyone with only a cursory introduction to AI programming" [1007, p. xxi]. One of these miniatures was *Micro Tale-Spin*, and indeed it was through this book chapter that Jim Meehan's system became the most famous early story generator (personal communication with Meehan, June 9, 2017). Later, in 1992, Warren Sack translated this version into a Common Lisp program [1073] that still runs today. Sack has recently written about this experience and about the phenomenon of the *miniature* [1074], and another recent paper by Nick Montfort and Natalia Fedorova draws on the topic [850]. My plan for this alternative dissertation was to provide historical case studies and coupled *reimplementations* for five forgotten systems in the history of computational media. In his paper that I have just referenced, Sack argues that reimplementation is a productive cultural form in the sense of humanities-style translation: "Whether we call these iterations 'rational reconstructions,' 'critical reimplementations,' 'translations,' or just 'rewrites,' it became clear to me why Donald Knuth employed the phrase 'the art of computer programming.' In programming, just as in art, copying is both the highest compliment and one of the most important forms of production" [1074, p. 14]. Recently, Montfort and collaborators have taken up the *Renderings* project, which takes on the challenge of coupled reimplementation *and* (natural-language) trans-

curred in tandem with my transition at UC Santa Cruz into the computational media department, from computer science, which reflected this emphasis on proceeding from media design to technology design, rather than vice versa.

In the process of attempting to devise a final media project to cap off my time in graduate school, I ended up revisiting the *Procedural Wobegon* design, which had now been shelved for over a year. This idea had stuck around in my head and was still very exciting to me, but it seemed that an experience with speech interaction was now out of scope. I decided to undertake the project, but I would reduce the scope to mere generative audio drama, not *interactive* generative audio drama.[8] In any event, no one has ever tried either, to my knowledge, so the project would still be ambitious (and novel).[9] With this plan in place, I began my exploration of integrating fine-grained character actions in *Talk of the Town*, which would eventually result in the *Hennepin* framework. I now had a simulation engine that could drive the generative audio drama—it would recount

lation of obscure non-English historical computational works [739]. One output of this project [852] was featured as part of the exhibition at least year's Workshop on the History of Expressive Systems [1047], which I co-organized with Mark Nelson; the exhibition also featured other reimplementations by Liza Daly [236] and Hugo van Kemenade [1289]. I hope to see growing interest in critical reimplementation, and in the history of computational media, more broadly. Maybe I will still write that book.

[8]Note: 'audio drama' is a more generalized term for what is often called 'radio drama', reflecting the emergence of examples that exist outside of the medium of radio, for instance, as podcasts [447]. Likewise, the term 'audio play' can be used to refer more generally to an individual instance of the form.

[9]A few projects have explored adjacent territory. Max Bense and Ludwig Harig's 1968 radio play *Der Monolog der Terry Jo* [97], an inspiration that I discuss below, incorporates procedurally generated text and synthesized speech (using a vocoder). *Breaking Out*, a 2012 audio play by BBC R&D, alters its content based on the time and geographic location of the listening session [189, 349]. To express such variation, a synthetic voice delivers some of the generated lines. While certainly related, neither of these antecedents generates the actual narrative of the audio play. To stretch a bit further, other related work includes *When in Rome* [888] and *Word Mine* [889], two voice-augmented tabletop games in Sensible Object's *Voice Originals* series, and Earplay's growing collection of voice-driven works of choice-based interactive storytelling: *Earplay Demo* [429], *Mr. Robot Daily Five Nine* [430], *Jurassic World Revealed* [292], *You and the Beanstalk* [295], *Office Hours* [293], *The Orpheus Device* [294], *Codename Cygnus Reactivated* [291], and the upcoming *Pugmire* [1320]. Though not efforts in generative audio drama, these are examples of audio-based computational media.

life in a *Hennepin* county—but I needed to design out the latter.

My original plan for the audio drama materialized after a conversation with my labmate Jacob Garbe. He had been advising me on prospective technical solutions for *speech synthesis* [584], which would be a central concern in the project. An initial challenge was that most of the synthetic voices at my disposal sounded robotic, and so I was considering how to lean into those aesthetics to design around them. Jacob suggested a brilliant idea, in the form of what Ian Horswill calls a *narrative alibi*. In his paper "Game Design for Classical AI", Horswill identifies design patterns that afford a clever circumnavigation, or even embrace, of the otherwise undesirable quirks inherent in procedural systems. One of these involves the deliberate crafting of a narrative framing that allows for the quirks:

> Narrative alibis[:] Use the game narrative to explain away the dysfunctional behavior. For example, make the AI characters be zombies, children, or kooky aliens. [506, p. 29]

Jacob's idea was to cleverly house the audio plays in a diegetic frame: in the distant future, a troupe of robot actors attempt to put on an episodic audio production concerning human life in an American county. This was a great narrative alibi, because it could explain away a number of potential quirks. If the voices sound robotic, it is because the actors are robots; if the procedurally generated language is awkward, it is because robots wrote it; if the events recounted do not make sense, it is because they express a robotical view of human life.[10]

I decided to run with this idea, with the working title *The Procedural Players Present*. Each episode would begin with the robot actors, the titular Procedural Players, being introduced with regard to the *Hennepin* characters they would be portraying in the episode—for example, "This week, we have *Matthew* as...

[10]The earlier procedural audio play *Breaking Out*, mentioned above, utilized this very conceit [447, pp. 104–105, 113].

Edward Runyan, the hard-working, hard-drinking mayor of Runyan, USA".[11] After introducing the full cast, a narrator would commence the episode with a phrase like, "And now, the Procedural Players present: 'A Nothing Place'". I excitedly began to sketch out some text snippets, which I synthesized using the various voices that are included in Amazon's *Polly* framework [1129]. To my dismay, the voices were far too robotic, even with the narrative alibi in place, and character dialogue was simply not aesthetically viable.

I considered abandoning the project at this point, because I was not willing to make such severe aesthetic concessions, but then I began experimenting with a particular *Polly* voice, called *Justin*. While *Justin* is designed to be coded as the voice of a male child, to me it sounded like a young woman. In particular, it reminded me of Joanna Newsom's evocative voice-over narration in Paul Thomas Anderson's *Inherent Vice* (2014) [41].[12] Working with a synthetic voice feels distinctly like a collaboration—craft text, or build a text generator, and literally hear the result—and this one got off to a good start. I began experimenting with prospective aesthetic styles, and eventually I converged on a lyrical approach to narration that was inspired by Nate DiMeo's delivery in the historiographic storytelling podcast *The Memory Palace* [266, 648]. By this point, I had converged on a workable aesthetic, and a journey from *Procedural Wobegon* to *The Procedural Players Present* to this ultimate form had culminated.

Sheldon County is a generative podcast that recounts the emergent characters and events of a a *Hennepin* simulated storyworld. More specifically, the project concerns the procedural generation of a multitude of podcast series, each of which recounts a unique storyworld for a specific listener's listening pleasure. To pro-

[11]Matthew is one of the Amazon *Polly* voices. My plan was to utilize all the English-trained voices with their names intact, and together they would compose the troupe of robot actors.

[12]Gimlet Creative producer Katelyn Bogucki independently made the same observation when she heard the voice later on (personal communication, May 25, 2018).

duce a given podcast series, a system called *Sheldon* sifts through the accumulated history of a *Hennepin* county to excavate promising material that it then narrativizes to construct episode scripts, which are then mounted in a podcast series through the utilization of speech synthesis and algorithmic music composition. As such, it is an archetypal work of *curationist emergent narrative*, and moreover one that features fully automatic curation. Unlike *Diol/Diel/Dial* and *Bad News*, the subjects of my earlier case studies, *Sheldon County* has not been completed. The project is currently in an early stage of development, but its entire architecture is fully developed and I have produced an end-to-end proof of concept in the form of two example generated episodes that have been released online.[13]

In this chapter, I will describe the design of *Sheldon County* and the architecture instantiated in its curation system, *Sheldon*. Because the project is still in development, it is too early to provide a full case study, which means this chapter will lack the 'pleasure' and 'pain' sections that characterized the structure of Chapters 8 and 10. However, it is already possible to discuss how the project, through its design and architecture, instantiates the curationist framework that is the subject of this dissertation. As such, I will discuss the project with regard to those concerns. Finally, though I have talked about the project in media appearances, this chapter represents its first official reporting.

12.1 Vision

Sheldon County is a generative podcast about life in a *Hennepin* simulated storyworld. More specifically, it is a collection of podcast series, each of which is procedurally generated, by a system named *Sheldon*, for a particular listener's listening pleasure. Both *Sheldon* and the eponymous county are named after Shel-

[13]These are available online at `https://soundcloud.com/james-ryan-887346009`.

don Klein, the unheralded pioneer of expressive artificial intelligence who featured heavily in Chapter 4. In this section, I will provide a brief description of how I imagine the experience itself working once it has been completed; a description of the actual architecture undergirding the experience appears in Section 12.3.

Inspiration: *Der Monolog der Terry Jo*

Before describing my vision for the *Sheldon County* experience, I would like to discuss an historical project that has inspired me in this effort. I mentioned above that there has been essentially no prior exploration into generative audio drama, but two projects have come close: while its narrative is not procedurally generated, BBC R&D's *Breaking Out* (2012), which I learned about only very recently, alters some of its content according to the geographic location of the listener and the time and date at which she listens. As cocreator Anthony Churnside explains in his dissertation on the project, the play's writer defined templates whose gaps would be filled in different ways depending on this data:

> The personalised variables identified are listed below:
>
> - Town where the story is set.
> - Three well known places located nearby where the story is set.
> - The weather at the time and place the listener is listening.
> - Films being shown at a cinema near to the listener?s location.
> - The date the listener is listening.
> - The news on the date the listener is listening.
> - The social network used by the listener.
>
> The variables which could be changed needed to be identified, and the writer needed to determine what the possible variables could be based on the location where the audience is listening. [189, p. 105]

The earlier example that has been particularly inspirational to me is *Der Monolog der Terry Jo* [97], a 1968 radio play by Max Bense—a pioneer of computer poetry and computational art more broadly [92, 1282]—and Ludwig Harig.

The piece recounts the true story of a young girl, Terry Jo Duperrault, who in 1961 was rescued after what initially appeared to be a yachting accident, but was in fact the scene of a puzzling multiple homicide [275]. For some time after being rescued, Terry Jo spoke in a highly peculiar way, as an effect of the trauma that she had endured aboard the sailing vessel. Eventually, she came out of this state and returned to speaking as she normally would, and this was the impetus for integrating computation into *Der Monolog der Terry Jo*. In his own dissertation, which in large part concerns this project, Kurt Beals explains its peculiar utilization of technology in this way:

> the radio play begins with a radical foray into the mechanical realm: when Terry Jo's voice is first heard, it is represented by a Vocoder, which reads sequences of letters or words randomly generated by a computer. Nine of these sequences are heard, with each successive sequence moving one step closer to natural language. In the first sequence, letters are selected completely at random, producing combinations such as "fyuiömgesevvrhykfds" [...]. In the second, letters are chosen using a stochastic process based on the letter frequency in a set of source texts: "h-rahhueber–sh-dfnupz." By the fourth, the frequency of three-letter combinations is taken into account, and already the text begins to look more like standard German: "has-wirklieb-stion-und-füsse-etwas." Finally, the ninth approximation uses words and not letters as the basic units, and takes into account the probability of any given three-word sequence, yielding phrases such as "das was weiss ist die sich niemals mit vater sagte er weg gehen." [91, p. 95][14]

The radio play was later adapted to the English language by Robert Goss, as *Monologue: Terry Jo*, and a production aired on Pacifica Radio in the Bay Area in 1984 [98]. I have heard both versions, and they are fantastic. What particularly inspires me is its creators' remarkably keen sense as to the aesthetic value of

[14]As Beals notes [91, pp. 79–83], this method evokes that which is employed in Claude Shannon's famous 1948 paper "A Mathematical Theory of Communication" [1137]. In this founding document of *information theory* [395], Shannon procedurally generates language using a statistical approach and mechanical means. Here, his first synthetic word is `xfoml`, which I have adopted as my Twitter handle.

procedural text and speech synthesis (in this case, vocoder speech). Bense and Harig did not incorporate these computational oddities merely because the result would sound strange and futuristic, but rather they took this approach to serve a larger artistic vision: in real life, Terry Jo fell into a state in which she could only speak in an alien cadence, but this gradually dissipated, and this is striking. In turn, Bense and Harig utilize procedural elements, and gradually reduce those elements over the course of the play, to parallel Terry Jo's linguistic fugue.

So what I take away from this example is a particular emphasis on using procedural text and speech synthesis—more broadly, procedural generation—not merely because the result is alien and strange and futuristic, but to satisfy particular artistic aims to which the technology is well-suited. In my case, I am trying to tell stories about computer characters living out little computer lives, and so the aesthetics of procedural text and speech synthesis meets my expressive aims. If I were to instead hand-author the episode text for voicing by a human narrator, there could be an aesthetic mismatch—the narration would sound too natural for the strange abstract world that it recounts. Incidentally, I have no choice but to generate text and use speech synthesis, since it would be impossible to handcraft narration that fits every possible world, but the fact that it is even worth doing in the first place is rooted in the concord between generative worlds and generative aesthetics. This is what I have learned from *Der Monolog der Terry Jo*.

Requesting a Series

A prospective listener navigates to the *Sheldon County* website to request that a unique podcast series be generated for her, which she does by entering in a *storyworld seed*. As I explained in Section 11.1.3, *Hennepin*'s procedure for world generation proceeds from a *random seed* [744], which means each possible storyworld is indexed by a unique number. In this way, in entering a particular storyworld seed, the prospective listener is in effect requesting the particular storyworld that is indexed by that number in *Hennepin*'s possibility space.[15] In addition to the random seed, the interface also requests contact information for the prospective listener (for example, an email address). Once she has entered both of these items, the interface expresses that her series will be ready in a day or so, at which point she will be contacted with information on how to access it for listening.

Creating a World

Once a new series has been requested, a cloud-based instance of the *Hennepin* framework kicks into gear to simulate the history of the requested storyworld, which it does by setting its storyworld seed to the one that has been requested. From here, the simulation proceeds according to all the procedures that were outlined in the previous chapter, resulting in a huge accumulation of data pertaining to that world and its history—in curationist terms, this is a *chronicle*. This chronicle will include something on the order of a few million transpired character actions, as well as miscellaneous data representing every entity that has ever existed and the relations among these entities (such as social networks).

Building a Series

From here, the system *Sheldon* proceeds to sift through this accumulation to excavate material pertaining to characters and events that could be recounted in the podcast series; this is *story sifting*. As I will explain in more detail in Section

[15]It is interesting to reconceptualize seeded generation as indexed retrieval. Rather than constructing an output by using the seed, the system simply retrieves it, fully intact, from the corresponding index in the possibility space.

12.3, *Sheldon* also does *narrativization* by using packages of the excavated material to construct actual episode scripts, for which narrational speech is synthesized using Amazon's *Polly* service [1129]. Additionally, *Sheldon* generates a musical score for each episode using the Google *Magenta* framework [297]. The result of this procedure is a set of audio files, each corresponding to one episode of the series. I am not sure yet how many episodes each series will comprise, but in any event *Sheldon* will generate an entire series all at once.

Packaging Up a Podcast

Once the episode files have been generated, as a final step *Sheldon* must package them up into an actual podcast. I have not yet developed an approach for this, but I imagine the files being uploaded to a server, where they would be associated with an *RSS feed* [446] that is created for the particular podcast series at hand. Each individual series will have a unique title that incorporates the storyworld seed—for example, *Sheldon County #1515459035*. At this point, a full-fledged podcast has been created, but my goal is to have the series be accessible from conventional podcast syndication platforms, such as the Apple *Podcasts* app.

As I argued in Section 4.2.5, it is critical to mount procedurally generated content in actual full-fledged media experiences. While I am targeting the genre of audio drama, I am more specifically targeting the medium of the podcast. We are in the "golden age of the podcast" [93, 475], and I want *Sheldon County* to work just like other podcasts work today—that means being available on the common platforms that listeners use to access them. Unfortunately, this could be challenging, since the syndication platforms make automatic targeting difficult (as a form of spam prevention and quality control). As a fallback approach, I could develop a standalone *Sheldon County* app that delivers the generated audio, but this would not quite meet my vision of mounting the curated stories in actual podcasts.

Listening

Finally, once the podcast becomes accessible, ideally via the popular syndication platforms, *Sheldon* contacts the listener, using the contact information that she provided in her request, to provide information on where and how to access her new podcast. Now she may listen to it, and potentially she could send her seed to a friend as a way of sharing her storyworld and its stories.

12.2 Background: *Expressionist*

In this section, I will provide a brief overview of the technology that I use in this project to drive the generation of episode scripts. *Expressionist* is a tool for building text generators that may be used in works of computational media, especially in cases where content should reflect dynamic aspects of an underlying system state. I have been developing the tool over the last few years with the help of several undergraduate collaborators: Taylor Owen-Milner, Max Fisher, Ethan Seither, and Tyler Brothers, the latter of whom is now leading its development. *Expressionist* has been used in a few videogame projects to date, as outlined in a series of case studies reported in an earlier paper [1049]. More recently, Jonathan Lessard and his *LabLabLab* collective have used it in *Hammurabi* (2017) [683], an ambitious videogame featuring what is to my knowledge the most extensive use of text generation in the medium to date. In a recent paper, Lessard and his collaborators outline their usage of *Expressionist* in the project [682]. Finally, the general approach here has been adapted for Spirit AI's *Character Engine* middleware technology, which to date has been used to produce a pair of in-house demonstration experiences [1088, p. 276].

Building a Grammar

Using *Expressionist*, an author defines a *context-free grammar* [149] whose nonterminal symbols may be attributed author-defined tags. As symbols are rewritten in the generation process, their tags are accumulated, and the resulting generated content comes packaged with all the tags of all the symbols that were rewritten. As such, rather than text alone, a *content bundle* coupling text and tags is produced. The use of tags is the tool's calling card and what distinguishes it from Kate Compton's *Tracery* [204, 205], a related tool that influenced its design. This tagging scheme works similarly to an *attribute grammar* [606, 607], except that tags are attached to nonterminal symbols rather than production rules.

Targeted Generation and Content Understanding

When text generation operates over a tagged grammar, two critical patterns are enabled: *targeted generation* and *content understanding*. While text generation using a context-free grammar conventionally entails a random sampling of the grammar's possibility space, the addition of tags provides for targeted generation: a system forms a *content request* that specifies the tags that should and should not be bundled with the generated content.[16] Moreover, because the approach entails the generation of content bundles, rather than text alone, a system can actually be made to "understand" the content. This is done by specifying system-side logic that processes the author-defined tags as a kind of structured data. For example, in my *Talk of the Town* dialogue generator [1045], a content bundle coupling generated text with the tags `move:greeting` and `tone:rude` is

[16] I call these, respectively, the *desired tags* and the *prohibited tags*.

understood by the system as being a rude greeting. When content understanding is possible, a system can update its state, in meaningful ways, in response to particular instances of generated content.

Productionist

To be clear, *Expressionist* is a web-based tool with which an author may define the source data that a text generator will operate over—in this case, a tagged grammar—but it is not itself a text generator. In the parlance of this project, a *Productionist* module is a text generator that one develops to meet the needs of a particular work of computational media. These modules vary across projects, because each puts into operation application-specific concerns that are captured in the tagset of the corresponding *Expressionist* grammar. Specifically, the job of a *Productionist* module is to satisfy content requests that are submitted to it by some other system, such as a dialogue manager (or *Sheldon*).

Reductionist

Additionally, there is another module that is built into the tool, called the *Reductionist*. This module is tasked with reducing the massive possibility space of a tagged grammar—where limited authoring may lead to trillions or quadrillions or more generable outputs—to a much smaller space of *expressible meanings*. An expressible meaning bundles a unique collection of grammar tags with all the *recipes* for generating outputs that come packaged with exactly that set of tags. The idea is that while a grammar may have a quadrillion paths—consider that each possible output equates to a unique path through the grammar—not all of the corresponding outputs will be meaningfully different with regard to the concerns of the project. For instance, subtle variations in sentence structure or lexical choice (such as those produced by synonym replacement) or even punctuation may create variant outputs that are technically distinct strings, but that really have the same essential meaning.

In *Expressionist*, this distinction is operational, since content packages that have the exact same tags will be understood as being the same by a system that requests them: by the pattern of content understanding, a system understands the generated content by reasoning over its bundled tags, since otherwise natural language processing would be required to understand the surface content. As such, each unique possible set of tags (that may appear in a content bundle) represents an individual meaning that a grammar can produce—hence 'expressible meaning'.

Once a set of expressible meanings has been produced by *Reductionist*, targeted generation becomes a trivial procedure: retrieve an expressible meaning associated with all of a content request's desired tags (and none of its prohibited tags), and then target any one of its recipes (since each produces the same tags). This is remarkable, because it recasts search over a space of *n*-illion grammar paths as search through on the order of a few thousand expressible meanings. While in an earlier project Joe Osborn and I explored the use of *software verification* techniques to carry out this work [907], I have since developed a different technique that is used in a *Reductionist* module that is built into *Expressionist* itself.[17] In addition to exporting a grammar, the tool also exports a file containing all of the expressible meanings and their associated recipes (again, these are grammar paths that produce text outputs that manifest the associated meanings).

[17]The term 'Reductionist' is Joe's coinage, which he came up with while exploring a technique for *reducing* tagged grammars to *symbolic visibly pushdown automata* [907].

Precondition Tags and Effect Tags

To express the power of *Expressionist* and to foreshadow aspects of the specific approach to text generation employed in *Sheldon County*, I would like to introduce a few critical authoring patterns that I have utilized in my own *Productionist* modules. By the first pattern, I define *precondition tags* that take the form of snippets of Python code that are evaluated by *Productionist*, at generation time, by applying them against the state of a storyworld. If a nonterminal symbol's precondition tags do not hold, then the symbol cannot be rewritten at that time (because the content it would produce would not be appropriate given the current state of the storyworld). Relatedly, I have also used precondition tags to gate symbol rewriting according to generation decisions that have already been made, as I will explain momentarily.

The second pattern concerns *effect tags*, which are specified as code snippets that update *Productionist*'s working memory when executed. In the course of a generation instance, the *Productionist* utilized by *Sheldon* maintains a *working memory*, in the style of a production system [245], which is a Python data structure that binds variables to data or to storyworld entities. For instance, the effect tag `m.side_character = m.arsonist.best_friend` would cause the working memory (bound to a variable `m`) to bind, to its variable `side_character`, the best friend of the character already bound to its variable `arsonist`. In this way, *Productionist* has all of a storyworld at its disposal.

The working memory also includes information about decisions that have already been made in the generation instance, including a copy of the partially generated episode script. This allows for the coordination of decisions within a generation instance—for instance, those pertaining to lexical choices—according to constraints that propagate from earlier symbol rewriting to guide later rewriting. In this way, backtracking is also supported, since a failed precondition on a later decision causes backtracking to an earlier choice point, where an alternative decision will be made instead. This authoring pattern relates to an interesting recent project by Eric Butler, Kristin Siu, and Alex Zook, in which they generate videogame boss behaviors by hacking generative grammars with similar authorial affordances [154]. Another related framework is Aarne Uotila's *Orcus* language [1283], with which an author can build a text generator that is driven by context-free grammars that maintain state within and across generation instances. More broadly, grammar-based text generation with backtracking is associated with the *chart generation* technique [572]. To be clear, by using precondition and effect tags, the attribute grammars of *Expressionist* are hacked to resemble something more akin to a production system.

Runtime Expressions

Finally, a third authoring pattern pertains to the use of what I call *runtime expressions*. When generating text for works of computational media, there will likely be cases where certain pieces of content cannot be known at authoring time, which means they cannot be included in an authored *Expressionist* grammar. For example, if an author defines a greeting generator for a roguelike game with generated characters, the names of the characters who will be greeted cannot be known at authoring time. As such, in these cases generated outputs will include gaps that must be filled in at runtime, where the unknown concerns will have been resolved. As such, a runtime expression is simply a snippet of code that can be evaluated to a string at runtime, as the content is being generated. Here is an example greeting with a runtime expression (enclosed in square brackets): `'Hi, [interlocutor.name].'`. Once this greeting is generated by a *Productionist*, the module can fill it in by evaluating the runtime expression, which will of course require the appropriate character to be bound to the variable `interlocutor` in the namespace of the generation procedure. When runtime expressions are used, the generable instances in an *Expressionist* grammar's possibility space are themselves *templated*, in the style of the authored procedural content in projects like *Prom Week* [803], *Versu* [326], and Curveship [847].

12.3 Podcast Generation

In this section, I will explain the operation of *Sheldon*, which is the system that handles podcast generation in the *Sheldon County* project. As I noted in Section 12.1, podcast generation is triggered when a prospective listener requests a podcast series to be generated for her, according to a particular storyworld seed. While I have produced two proof-of-concept episodes that were generated by an end-to-end prototype of this architecture (save for music generation), my approach is bound to change as the project developers further, and there is considerable authoring work that still needs to be undertaken. Namely, this work will entail the authoring of character actions in *Hennepin* and *episode spaces* in *Sheldon*, a notion that I will explain below. As such, to be clear, all of the forthcoming description is subject to change.

12.3.1 Storyworld

Upon a new series being requested, an instance of the *Hennepin* framework kicks into gear to simulate the history of a storyworld, which it does by setting its storyworld seed to the one associated with the request. In the version of *Hennepin* that I am using for this project, every county is hardcoded to have the name ‘Sheldon County’, the idea being that each is like an alternative version of the same essential setting. From here, the simulation proceeds according to all the procedures that were outlined in the previous chapter. Once the project is released, this computation will likely occur in the cloud, since that will be easier to set up than a dedicated server.

When the storyworld reaches a preset termination date—for example, October 18, 1987—*Hennepin* ceases its simulation of history. I imagine this will take several hours, and perhaps an entire day. At this point, a huge accumulation of data pertaining to that world and its history has been produced: on the order of a few million transpired character actions, as well as miscellaneous data representing every storyworld entity that has existed and various relations among these entities (such as social networks). In curationist terms—see Chapter 5—this data accumulation is a *chronicle*.

12.3.2 Nuggets

With the chronicle still in working memory, *Sheldon* commences with the curationist task of *story sifting*. In this project, that means sifting through the accumulated history of a *Hennepin* storyworld, as captured in the chronicle. The aim here is to excavate narratively potent material pertaining to characters and events that could be recounted in the podcast. Generally, this entails the employment of *sifting patterns*, in the style of the technical approach that I outlined in the opening of the Chapter 11. Of course, the point of excavating material is to construct a narrative artifact, which means that the patterns should also encode knowledge about how to recount the extracted material.

Narration Moves and Nuggets

For this reason, *Sheldon*'s sifting patterns are each associated with ideas for how the material may be used to build narrative artifacts. I call these descriptor tags *narration moves*, and each represents a high-level goal for a prospective episode that would be constructed out of that material (to recount the characters and events captured by the material); an example is `recount-arson-revenge`. Narration moves are similar to, for instance, the descriptors for plot fragments in *Universe* [666, 667] and the rhetorical goals of *Terminal Time* [768, 780, 268]. When a sifting pattern matches against material contained in the chronicle, it packages the extracted material into a *nugget* that also includes the narration moves associated with the sifting pattern. The aim here is to package the material in the nugget in a way that will make the narrativization task (about which more soon) more feasible.

An Example

Let us illustrate this with an example. In the current *Sheldon* codebase, a nugget is instantiated as an object of a class whose definition contains both the sifting pattern and a procedure for recording the extracted material in a structured way. As an example, here is a definition for a nugget class corresponding to the `arson-revenge` sifting pattern, which matches when a `set-fire` action has a causal ancestor—an earlier action on its causal chain—that is a `hatch-revenge-scheme` action with the same initiator:

```
class ArsonRevenge(object):
    """A sifting pattern for excavating events
        pertaining to revenge via arson."""

    def __init__(self, candidate):
        """Initialize an ArsonRevenge object."""
```

```
self.name = "arson-revenge"
self.narration_moves = [
    "recount-arson-revenge",
    "recount-revenge"
]
# Apply the pattern to the candidate to
# find out whether there's a match
self.match = (
    candidate.name == "set-fire" and
    candidate.find_ancestor(
        name="hatch-revenge-scheme",
        initiator=candidate.initiator
    )
)
if self.match:
    # Set nugget components
    self.set_fire = candidate
    # Retrieve "hatch-revenge-scheme" action
    # that precipitated the arson
    self.hatch_scheme = (
        candidate.find_ancestor(
            name="hatch-revenge-scheme",
            initiator=self.set_fire.initiator
        )
    )
    # Retrieve the arsonist
    self.arsonist = (
        self.hatch_scheme.binding("arsonist")
    )
    # Retrieve the character who was targeted
    # by the arson act
    self.target = (
        self.hatch_scheme.binding("target")
    )
    # Retrieve the scene of the arson
    self.scene = (
        self.set_fire.binding("building")
    )
    # Retrieve anyone killed in the fire
    self.casualties = (
        self.set_fire.all_bindings("casualty")
    )
```

```python
# Retrieve all actions that caused the
# hatching of the revenge scheme
self.lead_up = [
    a for a in self.hatch_scheme.ancestors
    if a.initiator is self.target
]
# Construct additional narration moves
self.narration_moves += [
    "explore-relationship({a}, {t})".format(
       a=self.arsonist.id,
       t=self.target.id
    ),
    "explore-relationship({t}, {a})".format(
       t=self.target.id,
       a=self.arsonist.id
    )
]
```

This nugget class is associated with an `arson-revenge` sifting pattern and four narration moves: `recount-arson-revenge`, `recount-revenge`, and the two `explore- relationship` variants, which I will explain momentarily.[18] The sifting pattern itself is captured by this snippet:

```python
# Apply the pattern to the candidate to
# find out whether there's a match
self.match = (
    candidate.name == "set-fire" and
    candidate.find_ancestor(
        name="hatch-revenge-scheme",
        initiator=candidate.initiator
    )
)
```

By this code, a match occurs when the candidate is a `set-fire` action that has a causal ancestor that is a `hatch-revenge-scheme` action with the same initiator. If the candidate triggers a match, then an actual structured nugget is built

[18] In the case of `explore-relationship`, the narration move accepts arguments (pertaining to the characters involved in the relationship).

using the code listed above. This entails retrieving the arsonist character, the target of her arson attempt, the scene of the crime, and any casualties who may have died in the fire. Additionally, the lead-up to the hatching of the scheme is retrieved. These will be a series of actions that were taken by the target and precipitated the revenge attempt by the arsonist, such as the `ridicule` actions shown in Figure 11.3. This kind of backward chaining from a ‘most reportable event’ (the `set-fire` action) to its causal ancestors is an operationalization of William Labov’s procedure for *narrative pre-construction* [632], which I outlined in Section 4.2.1. It is important to note that *Hennepin*’s extensive causal bookkeeping, reported in Section 11.2.7, makes story sifting feasible to do automatically.

12.3.3 Episode Spaces

Once a nugget has been extracted, it can be used to drive the curationist task of *narrativization*, which *Sheldon* also carries out. In this project, narrativization means constructing an episode script that recounts the material contained in the nugget. Such a script can be said to *perform* the narration moves that are associated with the nugget—thus, an `arson-revenge` may be used to construct an episode script that performs the narration moves `recount-arson-revenge`, `recount-revenge`, and the two `explore-relationship` variants.

Performing Narration Moves

Nuggets may have multiple narration moves as a way to provide multiple hooks into the chronicle for *Sheldon* to exploit. As I will explain shortly, in the process of generating a series of episodes, the system may propose narration moves to perform in future episodes, given what has been recounted in earlier ones. For example, a generated script may lead to a proposal for a future episode to explore the relationship of two characters that were mentioned in that script. Such a narration move would take a form like `explore-relationship(121, 309)`, where `121` and `309` are identifiers for the characters whose relationship would be explored. If a case of arson by revenge had transpired between these characters, that would certainly be an interesting way to explore their relationship, and this is why those narration moves are included in the example nugget definition above. In this way, nuggets can serve different kinds of storytelling goals, potentially simultaneously, which provides leverage to *Sheldon* and leads to diverse uses of the same patterns across different series. For example, an arson story could be the central event in one series and an interesting background episode in another.

Sheldon's narrativization procedure is structured as a series of attempts to perform particular narration moves. Once the system has sifted out as many nuggets as it can from the chronicle, given the sifting patterns that have been authored and the history of the particular storyworld at hand, it will have an initial set of candidate narration moves. As episode scripts are generated, additional narration moves may also be proposed, as I will explain momentarily. To perform a given narration move, a corresponding nugget must be at hand. If one is available, the move can be attempted by targeting a corresponding *episode space*. An episode space is an *Expressionist* grammar that makes use of the two authoring patterns that I described in Section 12.2: precondition tags and effect tags. To be precise, *Sheldon* employs a *Productionist* module that operates over the grammar to procedurally generate an episode script. When working memory is used in a *Productionist*, the procedure works more like executing a computer program than generating from a context-free grammar, and so I conceive of the episode spaces as computer programs for producing particular kinds of episodes.

Generating a Script

To generate a script out of an episode space, *Sheldon* forms a content request that includes (in its desired tags) a tag representing the targeted narration move. In my *Expressionist* grammar, I attach *move tags* to certain nonterminal symbols to express that the content produced by expanding them will lead to episode scripts that perform those narration moves. This allows for coordination between story sifting and narrativization: sifting patterns, nuggets, and episode spaces are all associated with narration moves, as a kind of *lingua franca*, which means sifting patterns can be used to extract nuggets that can be recounted by sampling an episode space. When the *Productionist* module receives a content request with the targeted narration move, it retrieves all of the expressible meanings associated with the corresponding move tag; these are called the *satisficing meanings*. Next, the module begins targeting recipes for those meanings, one by one. As I noted above, a recipe for an expressible meaning is a grammar path that may be traversed to produce content that expresses the meaning (as instantiated by the set of tags bundled with the content).

To target a recipe, the *Productionist* adds the contents of the nugget at hand into its working memory. Additionally, the working memory includes a pointer to the storyworld and to the entire chronicle, so that it may evaluate any precondition tags attached to nonterminal symbols in the episode space (since preconditions may reference anything about the storyworld or its history). Next, the module works through the grammar path specified in the recipe. If on that path the module encounters a nonterminal symbol with effect tags, it carries out the corresponding updates to its working memory. Alternatively, if it encounters a nonterminal symbol with precondition tags, it evaluates the preconditions against the working memory. If the preconditions fail, *Productionist* abandons the recipe and moves on to the next one, reverting the working memory to its earlier state; this is essentially a form of backtracking.

If all of an expressible meaning's recipes are exhausted, it moves on to the next satisficing meaning. If eventually it abandons the last satisficing meaning, then the targeted narration move cannot be performed given the nugget and storyworld at hand. As the *Productionist* is working through the recipe, it also updates the working memory to contain the partially generated episode script and a list of nonterminal symbols that have already been expanded. With this information recorded in the working memory, the system can coordinate decisions within a generation instance, so that certain later decisions cannot be made if incompatible ones were made earlier on. If at any point a recipe is successfully targeted (all of the preconditions hold for all of its nonterminal symbols), then the resulting generated output will be a complete episode script.

Title Proposals

When an episode script is generated, it comes bundled with three additional kinds of tags: *title proposals*, *narration proposals*, and *bible updates*. To be clear, these are simply examples of other kinds of tags that I attach to nonterminal symbols in authoring episode spaces. Each title proposal contains a prospective title for the episode, along with a *score* representing the utility of that title. Thus, title proposals accumulate as a byproduct of symbol rewriting, and some titles may be better to use than others. For example, if rewriting a particular nonterminal causes a very specific chunk of content to be included in the episode, a salient phrase in that chunk might make for the perfect title. In this case, the score in the corresponding title-proposal tag would be very high. Once a script has been generated, *Sheldon* selects a title probabilistically according to the distribution of scores across the set of proposals.

Narration Proposals

A narration proposal has the same structure, except that instead of titles it proposes narration moves for upcoming episodes. In this way, the specific content in a generated episode may suggest ideas for upcoming episodes, and so forth. For example, if a rivalry between two historical figures in the county's history was mentioned in one episode, it might be worthwhile to explore that rivalry in a follow-up episode. In this way, a `revenge-arson` nugget like the one above may be used to perform the `explore-relationship` narration move. More precisely, *Sheldon* would have already extracted the nugget (since the system extracts every possible nugget before generating the first episode), so it would simply be retrieved from an existing collection of extracted nuggets.

Bible Updates

Finally, *Sheldon* also maintains a *show bible*—a term adopted from television writing [318, 1324]—that contains critical information about characters and events that have already been recounted in previous episodes.[19] The bible-update tags are structured like effect tags, but they instead cause updates to the bible (and only after an episode script has successfully been generated). Precondition tags may also reference the contents of the show bible, which allows for coordination across particular authoring decisions made in each episode.

[19] A recent paper by Eugenio Concepción and collaborators proposes to operationalize the notion of a show bible in a brilliant shared evaluation task: a variety of story generators each operate from the same shared knowledgebase, meaning they produce stories about the same characters and events [206]. In this way, the outputs of different story generators would truly be comparable. (This kind of integration of heterogenous story generators reminds me of the *Slant* blackboard architecture [851] that I mentioned earlier on.) A curationist spin on this task would be to use, as the story bible, a chronicle that records the characters and events that actually emerged in a simulated storyworld. I hope their idea catches on.

12.3.4 Audio

Once an entire series worth of episode scripts has been generated, *Sheldon* must produce audio files for each of them. As I have mentioned above, speech synthesis in this project is handled by the Amazon *Polly* framework. This is a cloud-based service that handles requests and sends back audio files (in MP3 format). Requests take the form of the raw text that is to be voiced, but that text may also include expressions in the *Speech Synthesis Markup Language* (SSML) [1242, 62, 36], which allows for control over dimensions including the rate and pitch of the synthesized speech. As such, SSML tags can be utilized in authoring an episode space. Because *Polly* has a fairly low character limit, *Sheldon* breaks an episode script into smaller chunks, makes requests for each, and then stitches the resulting audio files together into a single file.

Score Generation

Finally, *Sheldon* will generate a musical score for each episode by utilizing the Google *Magenta* framework [297]. To be clear, this aspect of the architecture is only planned, not implemented. While my proof-of-concept episodes that are already available online did not feature generated scores—but rather music by the human composer Lee Rosevere—I have experimented with *Magenta* fairly extensively and reached a level of satisfaction with some of my trained models. *Magenta* is a essentially a wrapper around the *TensorFlow* framework [7] for *deep learning* [668] that is intended to fuel efforts in procedural generation (and also *creativity support tools* [1148]) in the domains of music and visual art. The models that I have been exploring apply *sequence-to-sequence* techniques [1222] to *MIDI* data [139]. MIDI is not audio, but instructions for producing a sequence of sounds, and so a critical final step for *Sheldon* will involve the synthesis of actual audio from the generated MIDI data. In my experimentation phase, I have been doing this manually using software instruments in an audio workstation, which means that I still need to settle on a technical solution for doing this automatically.[20]

Backup: Procedural Score Selection

If this approach to musical score generation does not pan out either technologically or aesthetically, another approach could entail the automatic selection of background music from a corpus of tagged audio files. This could be driven by *Expressionist* tags, for instance, with a special tagset that works like the title proposals, but instead proposes the mood of a background score according to particular decisions that have been made with regard to the content of an episode script. In this way, the accumulated mood proposals for a script could be used to programmatically retrieve an audio file for background music with the corresponding mood tags.

Letting *Bardo* Compose

A variant approach could follow recent work by Rafael Padovani and collaborators, who developed a system, called *Bardo*, that automatically scores a tabletop roleplaying game session by using speech recognition to sample the sentiment of the players, which it attempts to match by retrieving and playing an audio file with the corresponding mood [912]. Padovani's collaborator, Lucas Ferreira, is a fellow UC Santa Cruz PhD student who is now further developing *Bardo* as his dissertation project. His ultimate vision for the system is to algorithmically score real-time oral storytelling [337]. It would be interesting to apply this system to *Sheldon County*'s synthesized episode narration.

[20] While some do not appreciate *Magenta*'s bottom-up, knowledge-poor approach to music generation, I find it to be conceptually intriguing in the tradition of the *aleatoric music* [1020, 544], and earlier methods for *dice music* [471], as well as *chance art* [844] more broadly.

Generally, it would be feasible to also include, in the actual content of an episode space, special markers (tags salting the content) that could drive the operation of a musical score generator. This would be in the style of the SSML tags that modulate the synthesized speech, but instead these tags would send control signals to a score generator, for instance enabling musical transitions that are coupled with particular speech events in the generated narration audio. Indeed, this aspect of the project could easily become its own massive undertaking, and this is why I will likely take the easiest approach available, which at this time appears to be *Magenta*'s bottom-up method.

12.3.5 Delivery

Finally, as I have already explained in Section 12.1, *Sheldon*'s last task is to upload the generated audio files somewhere, associate them with a dedicated RSS feed, and contact the listener who requested the series to let her know that it has been created. Like I noted above, this aspect of the project has not been deeply explored at this early stage.

12.4 Curationism and *Sheldon County*

In this section, I will briefly summarize how *Bad News* instantiates the framework for curationist emergent narrative that I presented in Chapter 5. While my earlier collaborative project *Bad News* is itself a full-fledged curationist project, *Sheldon County* represents a first attempt at fully automatic curation. As I did in my earlier case studies on *Diol/Diel/Dial*, namely in Section 8.5, and *Bad News*, specifically in Section 10.6, I will now proceed to answer the following question: for each curationist architectural component, who or what in this project acts as

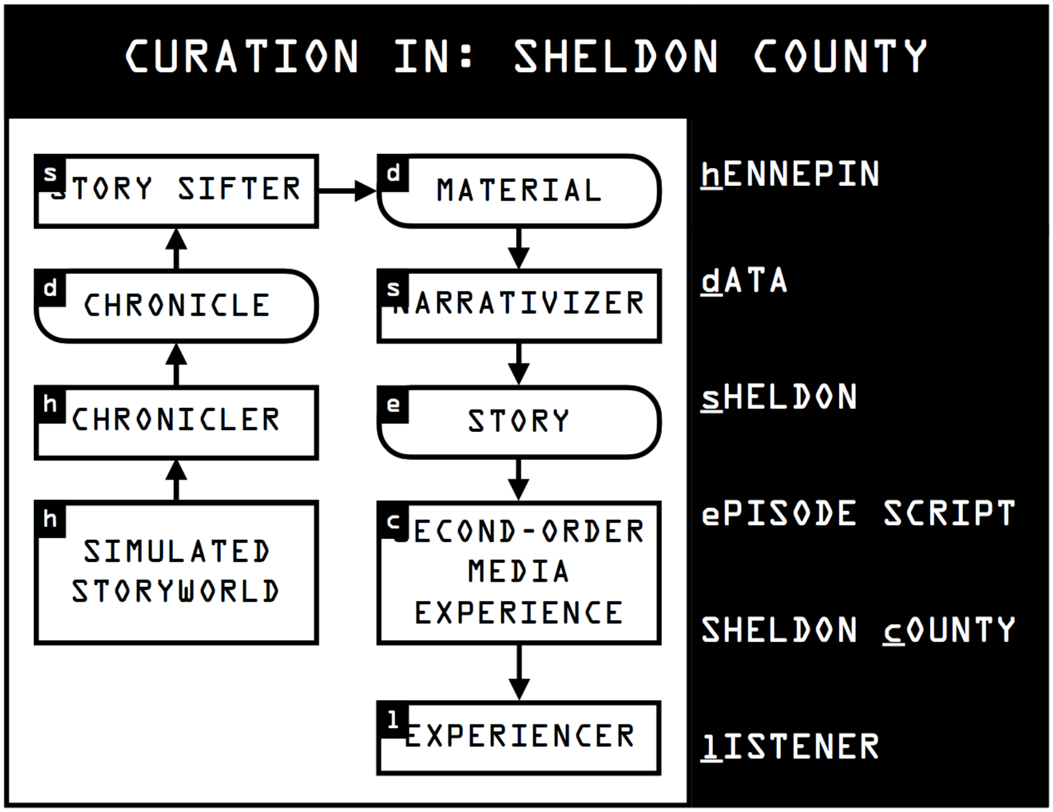


Figure 12.1: Curationist emergent narrative in *Sheldon County*. In this diagram, letter designators placed in the corner of each architectural component cue the corresponding entities in the *Sheldon County* project. Here, we find a variation on the *feedforward curation* pattern (shown above in Figure 5.2): the transpiring of a *Hennepin* storyworld is captured in a chronicle, which the system *Sheldon* sifts through to automatically excavate narratively potent raw material, which it then narrativizes to construct stories (episode scripts) that are ultimately mounted in a procedurally generated podcast series. While the earlier *Bad News* also instantiated the full curationist pipeline, it did so through the heavy utilization of human power. As such, *Sheldon County* represents an early exploration of fully automatic curationist emergent narrative.

that component? Figure 12.1 illustrates these concerns with regard to *Sheldon County*, which I will also outline in prose, component by component:

- **Experiencer.** *Sheldon County*'s *experiencer* is the podcast listener. Even though each series is generated specifically for one person, there could be potentially many listeners (and thus experiencers). This would occur when someone listens with friends, provides a seed to someone else, or shares audio files or instructions for accessing her podcast. In any case, a *Sheldon County* podcast is not an interactive experience, as Figure 12.1 shows.

- **Simulated storyworld.** A *Hennepin* simulation instance is the *simulated storyworld*. It is the procedurally generated American county that is recounted in a generated podcast series.

- **Chronicler.** As with *Diol/Diel/Dial* and *Bad News*, there is no distinct *chronicler* module at work in this project, but rather the associated functionality is built into the *Hennepin* codebase. To simplify, we can say that *Hennepin* is its own chronicler.

- **Chronicle.** A *chronicle* in this project is composed primarily of millions of character actions that have transpired in the course of a *Hennepin* storyworld's simulated history. Additionally, the record will contain other data pertaining to every entity that has existed during the span of that history, as well relations between those entities. For example, character social networks will be captured in the chronicle.

- **Story sifter.** Like *Bad News*, and unlike *Diol/Diel/Dial*, this project features *story sifting*, but critically this task is carried out automatically by the *Sheldon* system. As I explained in Section 12.3, *Sheldon* makes use of *sifting patterns* (first described in Section 5.4) to retrieve nuggets that bundle extracted material and associated narration moves. This aspect of *Sheldon* loosely computationalizes the manual story sifting that I carry out in my capacity as *Bad News* wizard, a process that I elucidated in Section 10.3.

- **Material.** As I just mentioned, *Sheldon*'s sifted *material* is packaged into *nuggets* that couple retrieved data with associated *narration moves*. These nuggets express the narrational value of the associated material in terms of what kind of stories could be told to recount it. This material takes the form of data stored in memory.

- **Narrativizer.** As in the case of story sifting, *Sheldon County* actually carries out *narrativization*—like *Bad News* and unlike *Diol/Diel/Dial*—but critically this is fully automatic. Narrativization in this system involves the utilization of *episode spaces*, which are authored *Expressionist* grammars with precondition tags that hook into the extracted material of nuggets, which allows for generated text that is tailored to the specific concerns captured therein. *Sheldon* is the narrativizer in this project, and the result of this procedure is a complete episode script.

- **Story.** In *Sheldon County*, narrativization results in a constructed narrative artifact—a *story*—that takes the form of a textual episode script. As I have argued at several points, namely in Section 4.2.5, such an artifact is not a full-fledged media experience. By the curationist framework, constructed narrative artifacts should be mounted in actual media experiences.

- **Media experience.** In curationism, the term 'media experience' specifically denotes the experience that constructed narrative artifacts (each being a 'story') are mounted into for human encounter. In *Sheldon County*, actual audio files (integrating speech and a musical score) are generated using the episode scripts, and the resulting collection of files is uploaded online and associated with an RSS feed. The result of that process is a podcast that has been procedurally generated for a particular listener's listening pleasure. This unique podcast series—something like *Sheldon County #1515459035*—

is in curationist terms a *media experience*.

Summary

To summarize, *Sheldon County* is a full-fledged work of curationist emergent narrative, and moreover one that makes use of fully automatic curation. Specifically, this project employs the *feedforward curation* variant of the curationist architecture (see Section 5.3.2), since it mounts stories that are constructed out of *Hennepin*'s emergent material into a second-order experience, a *Sheldon County* podcast. In its instantiation of this pattern, the system *Sheldon* works as a story sifter to excavate narratively potent raw material, which it then narrativizes to build stories that are ultimately mounted in procedurally generated podcast series. I am excited to develop the project further, and I hope to see other works that likewise explore automation of the curationist pipeline.

Chapter 13

Conclusion

I would like to conclude this earliest dissertation in computational media by first outlining the contributions that I have made in its course, with the hope that I have done well through these contributions to express the myriad concerns that are central in this nascent academic discipline. To wit, in this document I have made the following contributions:

- *A definition of emergent narrative.* After some deliberation, I converged on the following definition of 'emergent narrative' in Chapter 2: 'narrative that emerges out of computer simulation of character activity, or the methodology of generating narrative in that way'. My aim here was to join three distinct traditions that I view as instantiating the same form: the simulationist school of noninteractive story generation, as exemplified by systems like *Saga II*, Sheldon Klein's murder-mystery generator, and *Tale-Spin*; videogames driven by emergent character simulation, such as *Dwarf Fortress* and *The Sims*; and the bottom-up approach to academic interactive storytelling that is exemplified by the 'emergent narrative' research programs of Ruth Aylett, Sandy Louchart, Mariët Theune, Ivo Swartjes, and their respective collaborators. As such, by my ecumenical definition, emergent narrative may obtain in both interactive and noninteractive works, as long as uninhibited character simulation is at work.

- *An apologetics for emergent narrative.* While others have argued extensively about the merits of interactive emergent narrative, in Chapter 3 I worked to unpack the essence of character simulation itself. In doing so, I identified analogies of the form to nonfiction, lived experience, worldbuilding, and art brut. In particular, I agued at length that emergent narrative actually works more like nonfiction than fiction; here, my approach was to review a series of definitions of nonfiction, which I then applied to a work of curationist emergent narrative—Tim Dinee's *Oilfurnace* [255]—to show that the definitions hold for that work. Finally, I ended the chapter by identifying what I call *aesthetics of emergent narrative.*

- *A polemic against emergent narrative.* In addition to providing what is perhaps the most extensive defense of emergent narrative, I also contributed what may be the harshest critique of the form to date. In Chapter 4, I extended criticisms of emergent narrative advanced by various writers including Noah Wardrip-Fruin and Espen Aarseth to identify a series of pitfalls at the levels of both simulation and curation. Most primarily, I argued that simulation crafters often fail to enable emergent story structure, and that such structure can only truly manifest through a procedure of curation.

- *A refined approach to emergent narrative.* This contribution came in the form of my framework for curationist emergent narrative, which was the subject of Chapter 5 and of this dissertation more broadly. Here, I proceeded from the material contained in both my apologetics and polemic for the form—my challenge to myself was to devise a refined approach to emergent narrative that takes its harshest appraisals as an intellectual basis.

- *A set of artifacts, in the form of three simulation engines and three media experiences.* These contributions were constituted in my reporting of the completed simulation engines *World* and *Talk of the Town* in Chapters 7 and 9, the completed media experiences *Diol/Diel/Dial* and *Bad News* in Chapters 8 and 10, and the in-process engine *Hennepin* and the in-process experience *Sheldon County* in Chapters 11 and 12.

- *A set of case studies in curationist emergent narrative.* For each of the three reported media experiences (and in the corresponding chapters), I provided case studies that discussed them in terms of the curationist framework at the heart of this dissertation. In delivering these case studies, my goal was to provide an actionable account both of how curationism may be carried out and why that is worth doing. Toward this aim, I specifically identified a series of design patterns at the levels of both simulation and curation, including: contingent unlocking, causal bookkeeping, sifting patterns, and sifting heuristics. This contribution answers Ian Horswill's call for the utilization of the *case* as a reporting mechanism in computational media [505].

- *An autoethnography of my simulation and media practice.* Jonathan Lessard, another member of my reading committee, encouraged me to provide in this thesis an *autoethnography* recounting the evolution of my practice, and indeed this became one of its contributions. Across the openings of the chapters in Part II, I told the story of how my simulation and media practice, and the curationist framework, all coevolved over a six-year exploration of computation and research.

- *A history of emergent narrative.* While this dissertation was not an historical study, I did provide a significant historical account in the form of brief notes and mentions distributed across the document, especially in its footnotes. This included my tracing of the origins of the term 'emergent narrative' in Chapter 2; the first substantial account of the earliest known work of emergent narrative, *Saga II* (1960), in Chapter 4; an excavation of Sheldon Klein's murder-mystery generator, also in Chapter 4, which worked to show how the system has for decades been misunderstood (this was in the style of Noah Wardrip-Fruin's study of *Tale-Spin*); and numerous other details.

I would now like to articulate some brief thoughts on a notion that is bigger than curationist emergent narrative, but to which it still pertains: *generators that understand their outputs*. Earlier in this dissertation, I referenced researcher Mike Cook's pronouncement that "the aesthetics of big numbers is dead". In the original context of that death knell—it appeared in a *New Scientist* article about boredom with the videogame *No Man's Sky* [372]—writer Douglas Heaven also quotes researcher Gillian Smith:

> A vast computer-generated playground may instill awe, but size alone is not enough to sustain interest. "What actually gets us to engage with something is how well-crafted it is and how well it resonates," says Smith. [470, n.p.].

Around that same time, Kate Compton wrote an influential blog post identifying what she calls the *10,000 Bowls of Oatmeal problem*:

> So your algorithm may generate 18,446,744,073,709,551,616 planets. They may each be subtly different, but as they player is exploring them rapidly, will they be *perceived as different*? I like to call this problem the *10,000 Bowls of Oatmeal* problem. I can easily generate 10,000 bowls of plain oatmeal, with each oat being in a different position and different orientation, and *mathematically speaking* they will all be completely unique. But the user will likely just see *a lot of oatmeal. Perceptual uniqueness* is the real metric [201, p. 9]

In a jam-packed short paper on the topic, Rogelio Cardona-Rivera terms this the *Kaleidoscope Effect* and calls for the development of a poetics for *cognitively-grounded* procedural content generation [161]. Relatedly, Emily Short has advocated for generated content that is *meaningfully* correlated to a storyworld or to gameplay [1149]. Expounding on the issue himself, Mike Cook called for a more considered language of generative design [208], which Isaac Karth has now supplied in a brilliant paper on the *poetics of procedural generation* [569].

More broadly, in procedural generation, this decade has been marked by a growing emphasis on building generators that are more understandable to their designers [220, 211, 1381, 1133]—here, Gillian Smith and Jim Whitehead's 2010 paper "Analyzing the Expressive Range of a Level Generator" is the seminal work [1177].[1] Now, a new prospect is emerging: generators that understand themselves, or more specifically, their own outputs [1175]. This is the subject of Part III of Adam Summerville's dissertation [1213], which provides an array of techniques that may allow generators to more effectively reason about their own outputs.

In this document, I have introduced a framework, curationist emergent narrative, that suggests a related phenomenon: simulated storyworlds that understand themselves, so that they may tell their own stories. I have been calling for exploration of this notion since 2015, when in a conference paper I asked, "How does one make a system that can discern stories embedded in the morass of data that its simulation produces?" [1058, pp. 20–21]. That paper would ultimately become the seed for this dissertation, and with this document I believe that I have now answered the question to some extent. In articulating my curationist framework, and particularly its architecture, my aim has been to provide an actionable account of how to build systems that are able to tell stories about what happens

[1] In a clever inversion, Antonios Liapis and collaborators aim to make *designers more understandable to their generators* [688].

in simulated worlds. As this possibility becomes viable, an array of hitherto impossible media experiences are enabled, and the cutting edge of computational media is pushed forward.

But how viable *is* this possibility, at least right now? Let us turn to an excerpt from Jonathan Lessard's comments on an earlier draft of this thesis:

> To retrieve interesting stories, you will likely need to author story patterns or templates for the system to parse the chronicle: "a long lost treasure found", "revenge for humiliation", "the love triangle", etc. And the system will find them. And as is usually the case with templates, readers will soon see through them and go: "Oh, here goes the revenge for humiliation story, except with Bob instead of John"... How *really* different is that from a system that begins with these templates and then generates a story to fit it, the exact approach you contest throughout your dissertation? You will say: "the difference is that it *really* happened"... OK, but if I let a generator go until it gives me exactly what I wanted in the first place, how *real* is that?[2]

The general problem here is that a simulated storyworld cannot tell its own stories without harnessing some proceduralized method for doing so. As Jonathan indicated in his critique, one can define *sifting patterns* for retrieving certain kinds of material, but then the computer can only excavate material that matches those particular patterns. This is the impetus for *sifting heuristics*, as I explained earlier on, which is a notion that Jonathan connects to the historiographic practice:

> Historians don't approach their material looking for a pattern or a template. They often think in terms of "letting the archive speak"—that is keeping a mind open to serendipity. Of course, their *subjectivity* and *context* will inform what they find, but that is also what the "sifting patterns" miss. Since context and subjectivity change, different historians at different times from different places find different stories *in the same archives*![3]

[2]Personal communication, July 27, 2018. Adam Smith made a similar point at my thesis defense.

[3]Ibid.

Thus, for a computer to sift out material that enables not only a considerable variety of stories, but moreover ones that may surprise humans and faithfully express the *expressive range* [1177] of an underlying simulation, we will likely need to develop robust sifting heuristics. This is something that is not delivered in this thesis, because I do not personally know how to do this yet, and it is a very hard challenge. I would like to issue a call to others to roll up their sleeves and get to work on this critical problem in emergent narrative—indeed, sifting heuristics could likely be the topic of another entire dissertation.

Let us look then to a possible future, one in which an array of sifting heuristics have been developed and implemented in actual computational systems. In this future, would simulated worlds finally be able to tell their own stories?

Sadly, we encounter here a more nefarious problem, one that is potentially insurmountable—let us call it the *narrativization bottleneck*. Even if a system can employ sifting heuristics to excavate material that does not fit any defined pattern, but rather matches an abstract (and potentially novel) idea for a story, that is only half the battle. Once the material is extracted, there is still work to be done in order to produce an actual story—this is the task that I call *narrativization*. And it is here that we face a critical bottleneck: even if an essentially infinite diversity of stories can emerge in a simulated storyworld, and even if the system can employ sifting heuristics to extract a corresponding diversity of material, the computer will not be equipped with the means with which to construct actual stories out of all that material. Save for an automated system that can invent its own mechanisms for doing so, the existence of such methods will be predicated on human labor. But then we are back to the same fundamental problem: a world can only tell its own stories if a human has told it how. This is the narrativization bottleneck, and it is particularly devastating here because it works to constrain emergence, and the aesthetics of emergent narrative are diminished in turn.

How can we overcome the narrativization bottleneck? It would require something like the equivalent of sifting heuristics, but for the task of building actual stories out of raw material. While it is quite reasonable to imagine such an automated procedure in the domain of story sifting, it may not be within reason here. To be clear, this would require a computational procedure for telling *arbitrary* stories. While we find in the half century of work on *story generation* an array of methods for getting computers to tell stories, there is no system that can tell *arbitrary* stories. Story generators tell the stories that their human creators have built them to tell. To tell an arbitrary story is something that only humans can do, and in fact, doing so as a machine would likely require *artificial general intelligence* [397]. If there is a more tractable solution, I would love to hear about it, but this is a problem that could take a thousand dissertations to solve.

Sheldon Klein once wrote in a conference paper, "It is a rather long distance from the promise of the title to the fragmentary output just presented" [601, p. 30]. While this thesis makes a number of contributions—I have already enumerated them—I am not able to provide a solution to the narrativization bottleneck, and it is possible that no such solution will ever come to exist.

If that is the case, then what are we left with? Let me again quote from Jonathan Lessard's feedback on an earlier draft of this thesis:

> My own conclusion to your work is that the automatic curation and narration is likely to be an impasse (though I'm always open to be surprised) whereas we *know* that human curation can be delicious. We know this from Dwarf Fortress, we know this from Bad News, we know this from The Sims. Reading about your simulations was a real delight. I wanted to find amazing stories for myself like the ones you have chosen to narrate to us. And indeed, once I became proficient enough with Talk of the Town, I had a great time finding interesting stories for myself (who is this bartender who used to be a doctor after being a fireman???) However, these simulations are difficult to parse

> for oneself. But what if you provided us with curation affordances? Then *we* would find the stories that interest *us* and we would surprise *you* with what we find.

Regardless of the ultimate prospects for automatic curation, human-powered curation is not only feasible, but in fact it is already thoroughly demonstrated. As I argued at length in Chapter 5, it is how successful works of emergent narrative have worked all along. So how can we make it work even better? Incidentally, I think I have already answered that question.

All of the curationist steps leading up to narrativization do not have to be utilized as part of a fully automated pipeline. Alternatively, the corresponding modules could be framed as the components of a *curation support framework*: a computational system that is intended not to obviate human curation, but rather to *facilitate* it. In this way, even if arbitrary narrativization is impossible, we can still build full-fledged computational curationist systems, but ones in which the narrativization step is offloaded to human storytellers. In fact, this might actually be the most viable architectural pattern for curationist emergent narrative—perhaps you ought to try and find out.

Bibliography

[1] Can you think faster than this machine? [advertisement]. *Popular Science*, 168(2):16, Feb 1956.

[2] Exploring the fascinating world of language. *Business Machines*, XLVI(3):10–11, 1963. Courtesy IBM Corporate Archives.

[3] End note: Encyclopédie v.3. *Strategic Finance*, 90(3):64, Sep 2008.

[4] Espen Aarseth. Genre trouble. *Electronic Book Review*, 3, 2004.

[5] Espen Aarseth. A narrative theory of games. In *Proc. International Conference on the Foundations of Digital Games*, pages 129–133. ACM, 2012.

[6] Espen J Aarseth. *Cybertext: perspectives on ergodic literature*. Johns Hopkins University Press, 1997.

[7] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning. In *Proc. USENIX Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016.

[8] Robert P Abelson. The structure of belief systems. *Computer models of thought and language*, pages 287–339, 1973.

[9] Robert P Abelson and J Douglas Carroll. Computer simulation of individual belief systems. *American Behavioral Scientist*, 8(9):24–30, 1965.

[10] Andrew Adamatzky. *Game of Life Cellular Automata*. Springer, 2010.

[11] Ernest Adams. The challenge of the interactive movie. `http://www.designersnotebook.com/Lectures/Challenge/challenge.htm`, 1995.

[12] Ernest Adams. Interactive narratives revisited: Ten years of research. http://www.designersnotebook.com/Lectures/Interactive_Narratives_Revisit/interactive_narratives_revisited.htm, 2005.

[13] Ernest W Adams. *Fundamentals of Sports Game Design*. Pearson Education, 2014.

[14] Robert M Adams. The hobbit habit. *The New York Review of Books*, 24(19):22, Nov 1977.

[15] Tarn Adams. Simulation principles from Dwarf Fortress. In Steve Rabin, editor, *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, pages 519–521. CRC Press, 2015.

[16] Tarn Adams. Secret identities in Dwarf Fortress. In *Proc. Experimental AI in Games*, pages 22–25, 2017.

[17] Tarn Adams and Zach Adams. *Slaves to Armok: God of Blood Chapter II: Dwarf Fortress*. Bay 12 Games, 2002–.

[18] Nuno Afonso and Rui Prada. Agents that relate: Improving the social believability of non-player characters in role-playing games. In *Proc. International Conference on Entertainment Computing*, pages 34–45. Springer, 2008.

[19] Philip Agre. Toward a critical technical practice: Lessons learned in trying to reform ai. In Geoffrey C Bowker, Susan Leigh Star, William Turner, and Les Gasser, editors, *Social Science, Technical Systems and Cooperative Work: Beyond the Great Divide*. Psychology Press, 1997.

[20] Philip Agre and Ian Horswill. Lifeworld analysis. *Journal of Artificial Intelligence Research*, 6:111–145, 1997.

[21] Alexander Alberro and Blake Stimson. *Institutional critique: An anthology of artists' writings*. MIT Press, 2009.

[22] Anne-Mette Albrechtslund. Gender values in simulation games: sex and The Sims. In *Proc. Computer Ethics Philosophical Enquiries*, 2007.

[23] Leigh Alexander. Spector: Go emergent. *Gamasutra*, Nov 2013.

[24] Patricia A Alexander, Jonna M Kulikowich, and Sharon K Schulze. How subject-matter knowledge affects recall and interest. *American Educational Research Journal*, 31(2):313–337, 1994.

[25] Atocha Aliseda. *Abductive reasoning: Logical investigations into discovery and explanation*. Springer Science & Business Media, 2006.

[26] Mastress Alita. Sim stories: Hijinks from the world of *The Sims 3*. https://mastressalita.wordpress.com/. Last accessed: June 27, 2018.

[27] Mastress Alita. The stories and legacies index: A virtual library for all Sims stories and let's plays. https://storiesandlegaciesindex.wordpress.com. Last accessed: June 27, 2018.

[28] Jan M Allbeck. CAROSA: A tool for authoring npcs. In *Proc. International Conference on Motion in Games*, pages 182–193. Springer, 2010.

[29] Nicholas D Allen, John R Templon, Patrick Summerhays McNally, Larry Birnbaum, and Kristian J Hammond. Statsmonkey: A data-driven sports narrative writer. In *Proc. Computational Models of Narrative*, 2010.

[30] Thijs Alofs. The interactive storyteller a multi-user tabletop board game interface to support social interaction in ai-based interactive storytelling. Master's thesis, University of Twente, 2012.

[31] Thijs Alofs, Mariët Theune, and Ivo Swartjes. A tabletop board game interface for multi-user interaction with a storytelling system. In *International Conference on Intelligent Technologies for Interactive Entertainment*, pages 123–128. Springer, 2011.

[32] Thijs Alofs, Mariët Theune, and Ivo Swartjes. A tabletop interactive storytelling system: Designing for social interaction. *International Journal of Arts and Technology*, 8(3):188–211, 2015.

[33] M. Álvarez, Rafael Pérez y Pérez, and A. Aliseda. Grammar for prehispanic production: The case of el tajín style. In *Proc. International Joint Workshop on Computational Creativity*, pages 39–46, 2007.

[34] Sergio Alvarez-Napagao, Ignasi Gómez-Sebastià, Sofia Panagiotidi, Arturo Tejeda-Gómez, Luis Oliva, and Javier Vázquez-Salceda. Socially-aware emergent narrative. In *Proc. International Workshop on Agents for Educational Games and Simulations*, pages 139–150. Springer, 2011.

[35] Leila Amaral. The festive character of cyber art. *Technoetic Arts*, 8(3):255–265, 2011.

[36] Amazon. Speech Synthesis Markup Language (SSML) reference. https://developer.amazon.com/docs/custom-skills/speech-synthesis-markup-language-ssml-reference.html. Last accessed: June 26, 2018.

[37] Emmanuel Ameisen. Always start with a stupid model, no exceptions. https://blog.insightdatascience.com/always-start-with-a-stupid-model-no-exceptions-3a22314b9aaa, Mar 2018.

[38] Peter Bøgh Andersen. Multimedia phase-spaces. *Multimedia Tools and Applications*, 6(3):207–237, 1998.

[39] Peter Bøgh Andersen and Berit Holmqvist. Interactive fiction: artificial intelligence as a mode of sign production. *AI & Society*, 4(4):291–313, 1989.

[40] Peter Bøgh Andersen and Berit Holmqvist. Narrative computer systems: the dialectics of emotion and formalism. In Patrik O'Brian Holt and Noel Williams, editors, *Computers and Writing: State of the Art*, pages 40–67. Intellect Books, 1992.

[41] Paul Thomas Anderson. *Inherent Vice*. Warner Bros., 2014.

[42] Steve Anderson. Past indiscretions: Digital archives and recombinant history. In Marsha Kinder and Tara McPherson, editors, *Transmedia Frictions: The Digital, the Arts, and the Humanities*. University of California Press, 2007.

[43] Wes Anderson. *The Grand Budapest Hotel*. Fox Searchlight Pictures, 2014.

[44] Christopher Antoun, Matthew Antoun, James Owen Ryan, Ben Samuel, Reid Swanson, and Marilyn A Walker. Generating natural language retellings from Prom Week play traces. In *Proc. Procedural Content Generation in Games*, 2015.

[45] Rui Filipe Antunes and Frederic Fol Leymarie. Generative choreography: animating in real-time dancing avatars. In *Proc. International Conference on Evolutionary and Biologically Inspired Music and Art*, pages 1–10. Springer, 2012.

[46] Matthew A Appelbaum and Sheldon Klein. Meta-symbolic simulation system (MESSY) user manual (with forward: the history of MESSY). Technical Report TR272, University of Wisconsin, 1976.

[47] Thomas Aquinas. *Summa theologica*. 1702.

[48] Andrew Armour. Interactive fiction: Literature in the making. *The Geibun-Kenkyu: Journal of Arts and Letters)*, 48(3):1–14, 1986.

[49] Robert Audi. *Epistemology: A contemporary introduction to the theory of knowledge*. Routledge, 2005.

[50] Ruth Aylett. Narrative in virtual environments—towards emergent narrative. In *Proc. AAAI Fall Symposium on Narrative Intelligence*, pages 83–86, 1999.

[51] Ruth Aylett, Joao Dias, and Ana Paiva. An affectively driven planner for synthetic characters. In *Proc. International Conference on Automated Planning and Scheduling*, pages 2–10, 2006.

[52] Ruth Aylett and Sandy Louchart. I contain multitudes: creativity and emergent narrative. In *Proc. Creativity & Cognition*, pages 337–340. ACM, 2013.

[53] Ruth Aylett, Sandy Louchart, Joao Dias, Ana Paiva, Marco Vala, Sarah Woods, and Lynne Hall. Unscripted narrative for affectively driven characters. *Computer Graphics and Applications*, 26(3):42–52, 2006.

[54] Ruth S Aylett. Virtual Teletubbies: reapplying a robot architecture to virtual agents. In *Proc. Autonomous Agents*, pages 338–339. ACM, 1999.

[55] Ruth S Aylett, Sandy Louchart, Joao Dias, Ana Paiva, and Marco Vala. Fearnot!—an experiment in emergent narrative. In *Proc. International Workshop on Intelligent Virtual Agents*, pages 305–316. Springer, 2005.

[56] Sasha Azad and Chris Martens. Addressing the elephant in the room: Opinionated virtual characters. 2018.

[57] Sasha Azad, Jingyang Xu, Haining Yu, and Boyang Li. Scheduling live interactive narratives with mixed-integer linear programming. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, 2017.

[58] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.

[59] Baby Names Popularity. Namevoyager: Baby name wizard graph of most popular baby names – ryan. `http://www.babynamewizard.com/voyager#prefix=ryan&sw=both&exact=false`. Last accessed: December 2, 2018.

[60] Byung-Chull Bae, Yun-Gyung Cheong, and Robert Michael Young. Automated story generation with multiple internal focalization. In *Proc. Computational Intelligence and Games*, pages 211–218. IEEE, 2011.

[61] Andrew Baerg. Digital hoops history: NBA 2K12 and remediating basketball's past. *Communication & Sport*, 1(4):365–381, 2013.

[62] Paolo Baggia, Paul Bagshaw, Michael Bodell, De Zhi Huang, Lou Xiaoyan, Scott McGlashan, Jianhua Tao, Yan Jun, Hu Fang, Yongguo Kang, Helen Meng, Wang Xia, Xia Hairong, and Wu Zhiyong. Speech Synthesis Markup Language (SSML) version 1.1. `https://www.w3.org/TR/speech-synthesis11/`, Sep 2010.

[63] Richard P Bagozzi and Lynn W Phillips. Representing and testing organizational theories: A holistic construal. *Administrative Science Quarterly*, pages 459–489, 1982.

[64] Paul Bailey. Generating narrative text describing events and situations. Master's thesis, University of Edinburgh, 1996.

[65] Paul Bailey. A reader-based model of story generation—ph.d. thesis proposal. Technical Report DP181, University of Edinburgh, 1996.

[66] Paul Bailey. A reader-based model of story generation. In *Proc. Symposium on Artificial Intelligence and Creative Language: Stories and Humour*, 1999.

[67] Paul Bailey. Searching for storiness: Story-generation from a reader's perspective. In *AAAI Fall Symposium on Narrative Intelligence*, pages 156–163, 1999.

[68] Mikhail M Bakhtin. The dialogic imagination: Four essays. 1981.

[69] Mieke Bal and Eve Tavor. Notes on narrative embedding. *Poetics Today*, 2(2):41–59, 1981.

[70] Jean-Pierre Balpe. *Action poétique*. Number 95. 1984.

[71] Robert Balzer. A 15 year perspective on automatic programming. *IEEE Transactions on Software Engineering*, (11):1257–1268, 1985.

[72] Ann Banfield. A grammatical definition of the genre 'novel'. *Polyphonie–linguistique et littéraire*, (3):1–18, 2002.

[73] Woodrow Barfield. Intellectual property rights in virtual environments: considering the rights of owners, programmers and virtual avatars. *Akron Law Review*, 39:649, 2006.

[74] John Barger. IF, AI, and the confabulating-arranger model of interactive fiction. *Journal of Interactive Entertainment Design*, 7, 1994.

[75] Jorn Barger. 'The Policeman's Beard' was largely prefab. *Journal of Computer Game Design*, 6, Aug 1993.

[76] Camille Barot, Michael Branon, Rogelio E Cardona-Rivera, Markus Eger, Michelle Glatz, Nancy Green, James Mattice, Colin M Potts, Justus Robertson, Makiko Shukonobe, Laura Tateosian, Brandon Thorne, and R Michael Young. Bardic: Generating multimedia narrative reports for game logs. In *Proc. Intelligent Narrative Technologies*. 2017.

[77] Roland Barthes. The death of the author. *Aspen*, 5–6, 1967.

[78] Roland Barthes. *The rustle of language*. Macmillan, 1987.

[79] Roland Barthes and Lionel Duisit. An introduction to the structural analysis of narrative. *New Literary History*, 6(2):237–272, 1975.

[80] John Bateman. Sentence generation and systemic grammar: an introduction. *Iwanami Lecture Series: Language Sciences*, 1997.

[81] Joseph Bates. Computational drama in Oz. *ARTCOM*, 10(10), Dec 1990.

[82] Joseph Bates. Re: How do I get involved? `https://groups.google.com/d/msg/rec.arts.int-fiction/b_Y7oNX_WGU/b5lIfkAEWpYJ`, Apr 1990.

[83] Joseph Bates. The nature of characters in interactive worlds and the Oz project. Technical Report CMU-CS-92-200, Carnegie Mellon University, 1992.

[84] Joseph Bates. Virtual reality, art, and entertainment. *Presence: Teleoperators & Virtual Environments*, 1(1):133–138, 1992.

[85] Joseph Bates. The role of emotion in believable agents. *Communications of the ACM*, 37(7):122–125, 1994.

[86] Joseph Bates, A Bryan Loyall, and W Scott Reilly. An architecture for action, emotion, and social behavior. In *Proc. Modelling Autonomous Agents in a Multi-Agent World*, pages 55–68. Springer, 1992.

[87] Joseph Bates, A Bryan Loyall, and W Scott Reilly. Integrating reactivity, goals, and emotion in a broad agent. In *Proc. Cognitive Science Society*, 1992.

[88] Jean Baudrillard. *Simulacra and simulation*. University of Michigan Press, 1994.

[89] Claude Baum. *The system builders: The story of SDC*. System Development Corporation, 1981.

[90] Richard Bauman. *Story, performance, and event: Contextual studies of oral narrative*. Cambridge University Press, 1986.

[91] Kurt Beals. *From Dada to Digital: Experimental Poetry in the Media Age.* PhD thesis, University of California, Berkeley, 2013.

[92] Kurt Beals. "do the new poets think? it's possible": Computer poetry and cyborg subjectivity. *Configurations*, 26(2):149–177, 2018.

[93] Lene Bech. Is this the golden age of podcasts? *Columbia Journalism Review*, Nov 2014.

[94] Morteza Behrooz and Arnav Jhala. Modeling social interestingness in conversational stories. In *Proc. Australasian Computer Science Week Multiconference*, page 66. ACM, 2017.

[95] Morteza Behrooz, Afshin Mobramaein, Arnav Jhala, and Jim Whitehead. Cognitive and experiential interestingness in abstract visual narrative. In *Proc. Annual Meeting of the Cognitive Science Society*, 2018.

[96] Morteza Behrooz, Reid Swanson, and Arnav Jhala. Remember that time? telling interesting stories from past interactions. In *Proc. International Conference on Interactive Digital Storytelling*, pages 93–104. Springer, 2015.

[97] Max Bense and Ludwig Harig. *Der Monolog der Terry Jo.* Saarländischer Rundfunk, 1968.

[98] Max Bense, Ludwig Harig, and Robert Goss. *Monologue: Terry Jo.* Pacifica Radio, 1984.

[99] Stephen L Benton, Alice J Corkill, Janet M Sharp, Ronald G Downey, and Irina Khramtsova. Knowledge, interest, and narrative writing. *Journal of Educational Psychology*, 87(1):66, 1995.

[100] Peter L Berger and Thomas Luckmann. The social construction of reality: A treatise in the sociology of knowledge. 1967.

[101] Guy Bergstrom. Evil storytelling tricks no one should know. *The Red Pen of Doom*, Oct 2011.

[102] Edmund Callis Berkeley. Small robots: Report. `http://history-computer.com/Library/SmallRobots.pdf`, 1956.

[103] Lauren Berlant. On the case. *Critical Inquiry*, 33(4):663–672, 2007.

[104] Eric Berne. *Games people play: The psychology of human relationships.* Penguin Uk, 1968.

[105] Thomas Betts. *An Investigation of the Digital Sublime in Video Game Production*. PhD thesis, University of Huddersfield, 2014.

[106] Katrien Beuls and Luc Steels. Agent-based models of strategies for the emergence and evolution of grammatical agreement. *PloS one*, 8(3), 2013.

[107] Sebastian Hurup Bevensee and Henrik Schoenau-Fog. Conceptualizing productive interactivity in emergent narratives. In *Proc. International Conference on Interactive Digital Storytelling*, pages 61–64. Springer, 2013.

[108] Saroja Bhate and Subhash Kak. Pāṇini's grammar and computer science. *Annals of the Bhandarkar Oriental Research Institute*, 72(1/4):79–94, 1991.

[109] Christina Binkley. Minnesota's new cool image as 'the North'. *Wall Street Journal*, 2015.

[110] Robert I Binnick. An application of an extended generative semantic model of language to man–machine interaction. In *Proc. Computational Linguistics*, pages 1–34, 1969.

[111] Kim Binsted and Graeme Ritchie. Speculations on story puns. In *Proc. of the International Workshop on Computational Humour*, pages 151–159, 1996.

[112] Kim Binsted and Graeme Ritchie. Towards a model of story puns. *Humor*, 14(3):275–292, 2001.

[113] Johannes Birringer. Saira virous: Game choreography in multiplayer online performance spaces. In *Performance and Technology*, pages 43–59. Springer, 2006.

[114] Y Björnsson and T Marsland. Fritz 5.0 wins the 1997 Herschberg best-annotation award. *ICGA Journal*, 21(1):65–66, 1998.

[115] John B Black and Gordon H Bower. Story understanding as problem-solving. *Poetics*, 9(1-3):223–250, 1980.

[116] John B Black and Robert Wilensky. An evaluation of story grammars. *Cognitive Science*, 3(3):213–229, 1979.

[117] James Blackmon. Searle's wall. *Erkenntnis*, 78(1):109–117, 2013.

[118] Arthur F Blanchard. Movie-writer, 1916. US Patent 1,198,401.

[119] Leonard Bloomfield. *Language*. Henry Holt, 1933.

[120] Bruce Mitchell Blumberg. *Old tricks, new dogs: ethology and interactive creatures*. PhD thesis, Massachusetts Institute of Technology, 1997.

[121] Augusto Boal. *Theater of the Oppressed*. Pluto Press, 2000.

[122] Margaret A Boden. *Artificial intelligence and natural man*. MIT Press, 1977.

[123] Margaret A Boden. *The creative mind: Myths and mechanisms*. Psychology Press, 2004.

[124] Margaret A Boden. Foreword: How computational creativity began. In Tarek R Besold, Marco Schorlemmer, and Alan Smaill, editors, *Computational Creativity Research: Towards Creative Machines*. 2015.

[125] Maurizio Bolognini. The SMSMS project: Collective intelligence machines in the digital city. *Leonardo*, 37(2):147–149, Apr 2004.

[126] Stephanie Boluk and Patrick LeMieux. Dwarven epitaphs: Procedural histories in Dwarf Fortress. In N Katherine Hayles and Jessica Pressman, editors, *Comparative Textual Media: Transforming the Humanities in the Postprint Era*, pages 125–154. University of Minnesota Press, 2013.

[127] Stephanie Boluk and Patrick LeMieux. *Metagaming: Playing, competing, spectating, cheating, trading, making, and breaking videogames*. University of Minnesota Press, 2017.

[128] Jorge Luis Borges. *Labyrinths: Selected stories & other writings*. New Directions, 1964.

[129] Edwin G Boring. Dual role of the zeitgeist in scientific creativity. *The Scientific Monthly*, 80(2):101–106, 1955.

[130] Daniel Bormann and Tobias Greitemeyer. Immersed in virtual worlds and minds: effects of in-game storytelling on immersion, need satisfaction, and affective theory of mind. *Social Psychological and Personality Science*, 6(6):646–652, 2015.

[131] Edith Borroff and J Bunker Clark. *American operas: a checklist*. Harmonie Park Press, 1992.

[132] Nick Bostrom and Eliezer Yudkowsky. The ethics of artificial intelligence. *The Cambridge Handbook of Artificial Intelligence*, pages 316–334, 2014.

[133] Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, and Leo Wanner. Perspective-oriented generation of football match summaries: Old tasks, new challenges. *ACM Transactions on Speech and Language Processing*, 9(2):3, 2012.

[134] John Boudreau. A romance novel with byte: Author teams up with computer to write book in steamy style of Jacqueline Susann. *Los Angeles Times*, page E6, Aug 1993.

[135] Kevin K Bowden, Grace I Lin, Lena I Reed, Jean E Fox Tree, and Marilyn A Walker. M2D: Monolog to dialog generation for conversational story telling. In *Proc. International Conference on Interactive Digital Storytelling*, pages 12–24. Springer, 2016.

[136] Kevin K Bowden, Jiaqi Wu, Shereen Oraby, Amita Misra, and Marilyn Walker. Slugbot: An application of a novel and scalable open domain socialbot framework. *arXiv preprint arXiv:1801.01531*, 2018.

[137] Mattie Brice. Top 10 fav games of 2016. `http://www.mattiebrice.com/top-10-fav-games-of-2016`, Dec 2016.

[138] Selmer Bringsjord and Dave Ferrucci. BRUTUS and the narrational case against Church's thesis. In *Proc. Narrative Intelligence Symposium*, 1999.

[139] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep learning techniques for music generation—a survey. Technical report, arXiv, 2017.

[140] Bruce K Britton. What makes stories interesting. *Behavioral and Brain Sciences*, 6(4):596–597, 1983.

[141] Mark Brokington. Building a reputation system: Hatred, forgiveness, and surrender in neverwinter nights. *Massive Multiplayer Game Development*, 1:454–463, 2003.

[142] Cyril Brom, Michal Bída, Jakub Gemrot, Rudolf Kadlec, and Tomáš Plch. Emohawk: searching for a âĂIJgoodâĂİ emergent narrative. In *Proc. Joint International Conference on Interactive Digital Storytelling*, pages 86–91. Springer, 2009.

[143] Allan G Bromley. Charles Babbage's analytical engine, 1838. *Annals of the History of Computing*, 4(3):196–217, 1982.

[144] Rodney A Brooks. Elephants don't play chess. *Robotics and autonomous systems*, 6(1-2):3–15, 1990.

[145] John Seely Brown, Richard R Burton, Alan G Bell, and Robert J Bobrow. SOPHIE: A sophisticated instructional environment. Technical report, 1974.

[146] Matt Brown. The power of projection and mass hallucination: Practical AI in The Sims 2 and beyond, 2006. Presented at Artificial Intelligence in Interactive Digital Entertainment.

[147] Miles Brundage. Artificial intelligence and responsible innovation. In *Fundamental Issues of Artificial Intelligence*, pages 543–554. Springer, 2016.

[148] Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitzoff, Bobby Filar, et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. Technical Report 1802.07228, arXiv, 2018.

[149] Alan Bundy and Lincoln Wallen. Context-free grammar. In *Catalogue of Artificial Intelligence Tools*, pages 22–23. Springer, 1984.

[150] Robin Burkinshaw. Alice and Kev. `https://aliceandkev.wordpress.com/`, 2009. Last accessed: May 22, 2018.

[151] Alex Burns. Civilization III: Digital game-based learning and macrohistory simulations. *Australian Foresight Institute/Disinformation*, 2002.

[152] Lauren Burr. Bicycles, bonfires and an airport apocalypse: The poetics and ethics of netprov. *Hyperrhiz: New Media Cultures*, (11), 2015.

[153] William S Burroughs. The cut-up method of Brion Gysin. In *The Third Mind*, pages 29–33. 1961.

[154] Eric Butler, Kristin Siu, and Alexander Zook. Program synthesis as a generative method. In *Proc. International Conference on the Foundations of Digital Games*, page 6. ACM, 2017.

[155] Charles B Callaway and James C Lester. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252, 2002.

[156] P Michael Campbell. Interactive fiction and narrative theory: Towards an anti-theory. *New England Review and Bread Loaf Quarterly*, 10(1):76–84, 1987.

[157] Nicolas Campion, Daniel Martins, and Alice Wilhelm. Contradictions and predictions: Two sources of uncertainty that raise the cognitive interest of readers. *Discourse Processes*, 46(4):341–368, 2009.

[158] Rosa Carbo-Mascarell. Walking simulators: The digitisation of an aesthetic practice. In *Proc. International Joint Conference of DiGRA and FDG*, 2016.

[159] Beth Cardier. *Unputdownable: How the agencies of compelling story assembly can be modelled using formalisable methods from Knowledge Representation, and in a fictional tale about seduction.* PhD thesis, University of Melbourne, 2012.

[160] Roger Cardinal. Outsider art and the autistic creator. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1522):1459–1466, 2009.

[161] Rogelio E Cardona-Rivera. Cognitively-grounded procedural content generation. In *Proc. AAAI Workshop on What's Next for AI in Games*, pages 1027–1028, 2017.

[162] Rogelio E Cardona-Rivera and Boyang Li. PLOTSHOT: Generating discourse-constrained stories around photos. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, 2016.

[163] Felice Cardone and J Roger Hindley. History of lambda-calculus and combinatory logic. *Handbook of the History of Logic*, 5:723–817, 2006.

[164] Don Carson. Environmental storytelling: Creating immersive 3D worlds using lessons learned from the theme park industry. *Gamasutra*, 2000.

[165] Don Carson. Environmental storytelling, part ii: Bringing theme park environment design techniques to the virtual world. *Gamasutra*, 2000.

[166] Marcus Carter. Emitexts and paratexts: Propaganda in EVE Online. *Games and Culture*, 10(4):311–342, 2015.

[167] Marcus Carter, Kelly Bergstrom, and Darryl Woodford. *Internet Spaceships Are Serious Business: An EVE Online Reader*. University of Minnesota Press, 2016.

[168] David B Carvalho, CT Pozzer, EG Clua, and EB Passos. A perception simulation architecture for plot generation of emergent storytelling. In *Proc. Computer Games, Multimedia and Allied Technology*, pages 6–11, 2012.

[169] Luigi Luca Cavalli-Sforza and Marcus W Feldman. *Cultural transmission and evolution: A quantitative approach*. Princeton University Press, 1981.

[170] Marc Cavazza, Fred Charles, and Steven J Mead. Character-based interactive storytelling. *Intelligent Systems*, 17(4):17–24, 2002.

[171] Central Opera Service. *Directory of contemporary operas & music theater works & North American premiers (1980–1989)*. 1990.

[172] Alan Chamberlain, Mads Bødker, Maria Kallionpää, Richard Ramchurn, David De Roure, Steve Benford, Alan Dix, and Hans-Peter Gasselseder. The design of future music technologies:âĂŸsounding outâĂŹai, immersive experiences & brain controlled interfaces. In *Proc. Audio Mostly*. ACM, 2018.

[173] Raymond Chandler. *The simple art of murder.* Vintage Crime/Black Lizard, 2002 [1950].

[174] Alenda Chang, Jesús Costantino, and Braxton Soderman. The multiple lives of permadeath: An introduction. *Journal of Gaming & Virtual Worlds*, 9(2):103–121, 2017.

[175] Hsueh-Min Chang and Von-Wun Soo. Planning-based narrative generation in simulated game universes. *Transactions on Computational Intelligence and AI in Games*, 1(3):200–213, 2009.

[176] Heather Chaplin and Aaron Ruby. *Smartbomb: The quest for art, entertainment, and big bucks in the videogame revolution.* Algonquin Books, 2006.

[177] Seymour Benjamin Chatman. *Story and discourse: Narrative structure in fiction and film.* Cornell University Press, 1980.

[178] Kenneth Chen. Pleasure Island: An instruction booklet for a new and virtual life. *CTHEORY*, Jun 2000.

[179] Kenneth Chen and Stefan Rank. Towards design principles for humor in interactive emergent narrative. In *Proc. Digital Games Research Association*, 2018.

[180] Sherol Chen, Mark J Nelson, Anne Sullivan, and Michael Mateas. Evaluating the authorial leverage of drama management. In *Proc. Intelligent Narrative Technologies*, pages 20–23, 2009.

[181] Yun-Gyung Cheong, Arnav Jhala, Byung-Chull Bae, and Robert Michael Young. Automatically generating summary visualizations from game logs. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, pages 167–172, 2008.

[182] Yun Gyung Cheong and R Michael Young. Computational model of focalization in narrative planning. In *Proc. International and Interdisciplinary Symposium of the Narratology Research Group at the University of Hamburg*, 2006.

[183] Yun-Gyung Cheong and Robert Michael Young. A framework for summarizing game experiences as narratives. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, pages 106–108, 2006.

[184] Ria Cheyne. Created languages in science fiction. *Science Fiction Studies*, pages 386–403, 2008.

[185] Hui-Hsien Chou and James A Reggia. Emergence of self-replicating structures in a cellular automata space. *Physica D: Nonlinear Phenomena*, 110(3-4):252–276, 1997.

[186] David Christian. Strategic deception in agents. Master's thesis, North Carolina State University, 2004.

[187] Alonzo Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, pages 346–366, 1932.

[188] Alonzo Church. Application of recursive arithmetic to the problem of circuit synthesis. Technical report, Institute for Defense Analysis, Princeton, 1960.

[189] Anthony WP Churnside. *Object-based radio: Effects on production and audience experience*. PhD thesis, The University of Salford, 2016.

[190] Eric Clarke. *Lo-Fi aesthetics in popular music discourse*. PhD thesis, University of Oxford, 2014.

[191] Ralph D Clifford. Intellectual property in the era of the creative computer program: Will the true creator please stand up. *Tulane Law Review*, 71:1675, 1996.

[192] John H Clippinger. *A Discourse Speaking Program As A Preliminary Theory Of Discourse Behavior And A Limited Theory Of Psychoanalytic Discourse*. PhD thesis, University of Pennsylvania, 1975.

[193] John H Clippinger. Speaking with many tongues: Some problems in modeling speakers of actual discourse. In *Proc. Workshop on Theoretical Issues in Natural Language Processing*, pages 68–73. Association for Computational Linguistics, 1975.

[194] William F Clocksin and Christopher S Mellish. *Programming in Prolog*. Springer Science & Business Media, 2003.

[195] Deborah A Cobb-Clark and Stefanie Schurer. The stability of big-five personality traits. *Economics Letters*, 115(1):11–15, 2012.

[196] Dorrit Cohn. *The distinction of fiction*. Johns Hopkins University Press, 2000.

[197] Jean-Philippe Cointet and Camille Roth. How realistic should knowledge diffusion models be? *Journal of Artificial Societies and Social Simulation*, 10(3):5, 2007.

[198] Kenneth Mark Colby. *Artificial paranoia: A computer simulation of paranoid processes*. Elsevier, 2013.

[199] Michael Collins. Discriminative reranking for natural language parsing. In *Proc. International Conference on Machine Learning*, 2000.

[200] Nick Collins. Live coding of consequence. *Leonardo*, 44(3):207–211, 2011.

[201] Kate Compton. So you want to build a generator. `http://www.galaxykate.com/buildagenerator-kcompton.pdf`, Feb 2016.

[202] Kate Compton. Kambamanx. `http://tracery.io/kambamanx/`, 2017. Last accessed: December 14, 2018.

[203] Kate Compton. Little procedural people: playing politics with generators. In *Proc. Procedural Content Generation*. ACM, 2017.

[204] Kate Compton, Benjamin Filstrup, and Michael Mateas. Tracery: Approachable story grammar authoring for casual users. In *Proc. Intelligent Narrative Technologies*, 2014.

[205] Kate Compton, Ben Kybartas, and Michael Mateas. Tracery: an author-focused generative text tool. In *Proc. International Conference on Interactive Digital Storytelling*, pages 154–161. Springer, 2015.

[206] Eugenio Concepción, Gonzalo Méndez, Pablo Gervás, and Carlos León. A challenge proposal for narrative generation using CNLs. In *Proc. International Natural Language Generation Conference*, pages 171–173, 2016.

[207] John Conway. The game of life. *Scientific American*, 223(4):4, 1970.

[208] Michael Cook. Alien languages: How we talk about procedural generation. *Gamasutra*, Aug 2016.

[209] Michael Cook. Ethical procedural generation. In Tanya X. Short and Tarn Adams, editors, *Procedural Generation in Game Design*, pages 43–54. AK Peters/CRC Press, 2017.

[210] Michael Cook, Mirjam P Eladhari, Adam M Smith, Gillian Smith, Tommy Thompson, and Julian Togelius. AI-based games: Contrabot and What Did You Do? In *Proc. International Conference on the Foundations of Digital Games*, 2015.

[211] Michael Cook, Jeremy Gow, and Simon Colton. Danesh: Helping bridge the gap between procedural generators and their output. In *Proc. Procedural Content Generation*, 2016.

[212] William Wallace Cook. *Plotto: a new method of plot suggestion for writers of creative fiction*. Ellis publishing company, 1928.

[213] Amra Copcic, Sophie McKenzie, and Michael Hobbs. Permadeath: A review of literature. In *Proc. IEEE International Games Innovation Conference*, pages 40–47. IEEE, 2013.

[214] Alfred Correira. Computing story trees. *Computational Linguistics*, 6(3-4):135–149, 1980.

[215] Mario Costa. *Le sublime technologique*. Iderive, 1994.

[216] Paul T Costa and Robert R McCrea. *Revised NEO personality inventory (NEO PI-R) and NEO five-factor inventory (PNEO-FFI)*. Psychological Assessment Resources, 1992.

[217] William V Costanzo. Reading interactive fiction: implications of a new literary genre. *Educational Technology*, 26(6):31–35, 1986.

[218] Jasmine Cox, Matthew Brooks, Ian Forrester, and Mike Armstrong. Moving object-based media production from one-off examples to scalable workflows. *Motion Imaging*, 127(4):32–37, 2018.

[219] Kate Cox. Dwarf Fortress: The game that takes a 238-page illustrated technical manual to learn. *Kotaku*, Jun 2012.

[220] Rui Craveirinha, Lucas Santos, and Licínio Roque. An author-centric approach to procedural content generation. In *Advances in Computer Entertainment*, pages 14–28. Springer, 2013.

[221] Chris Crawford. *Trust & Betrayal: The Legacy of Siboot*. Mindscape, 1987.

[222] Chris Crawford. Assumptions underlying the erasmatron interactive storytelling engine. In *AAAI Fall Symposium on Narrative Intelligence*, pages 112–114, 1999.

[223] Chris Crawford. Deikto: A language for interactive storytelling. In Pat Harrigan and Noah Wardrip-Fruin, editors, *Second Person*. MIT Press, 2007.

[224] Chris Crawford. *Chris Crawford on Interactive Storytelling*. New Riders, 2012.

[225] Luka Crnkovic-Friis and Louise Crnkovic-Friis. Generative choreography using deep learning. In *Proc. International Conference on Computational Creativity*, 2016.

[226] William Croft. *Explaining language change: An evolutionary approach.* Pearson Education, 2000.

[227] Jason James Crouch. *Mediated messages: constructions of intimate communication through the use of digital technologies, and the extent to which such encounters can be conceptualised as one-to-one performance.* PhD thesis, Manchester Metropolitan University, 2017.

[228] Carolina Cruz-Neira. *Virtual reality based on multiple projection screens: the CAVE and its applications to computational science and engineering.* PhD thesis, University of Illinois at Chicago, 1995.

[229] Carolina Cruz-Neira, Daniel J Sandin, Thomas A DeFanti, Robert V Kenyon, and John C Hart. The CAVE: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–72, 1992.

[230] David Cummiskey. Reference failure and scientific realism: a response to the meta-induction. *The British journal for the philosophy of science*, 43(1):21–40, 1992.

[231] Gregory Currie. What is fiction? *The Journal of Aesthetics and Art Criticism*, 43(4):385–392, 1985.

[232] Maria Cutumisu and Duane Szafron. An architecture for game behavior ai: Behavior multi-queues. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, pages 20–27, 2009.

[233] Barbara Czarniawska. *Narratives in social science research.* Sage, 2004.

[234] Jedrzej Czarnota. Do EVE Online players dream of Icelandic spaceships? *Internet Spaceships Are Serious Business: An EVE Online Reader*, 2016.

[235] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of Oz studies—why and how. *Knowledge-based systems*, 6(4):258–266, 1993.

[236] Liza Daly. SAGA III: A creative reimplementation of SAGA II, created by Douglas T. Ross and Harrison R. Morse. In *Proc. History of Expressive Systems*, to appear.

[237] Rossana Damiano, Vincenzo Lombardo, and Antonio Pizzo. DoPPioGioco: Playing with the audience in an interactive storytelling platform. In *Proc. Complex, Intelligent, and Software Intensive Systems*, pages 287–298. Springer, 2017.

[238] Rossana Damiano, Vincenzo Lombardo, and Antonio Pizzo. Thinning the fourth wall with intelligent prompt. In *Proc. International Conference on Interactive Digital Storytelling*, pages 206–218. Springer, 2017.

[239] Arthur C Danto. Narrative sentences. *History and Theory*, 2(2):146–179, 1962.

[240] Kerstin Dautenhahn. Embodied cognition in animals and artifacts. *Proc. Embodied Cognition and Action*, pages 27–32, 1996.

[241] Anthony Davey. *The formalisation of discourse production*. PhD thesis, University of Edinburgh, 1974.

[242] Anthony Davey. *Discourse production: a computer model of some aspects of a speaker*. University Press, 1978.

[243] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.

[244] Justin Davis. Justin Davis's favorite games of all time. *IGN*, Feb 2014.

[245] Randall Davis and Jonathan King. An overview of production systems. Technical Report STAN-CS75-524, Stanford University, 1975.

[246] Michael Dawson. *Python programming for the absolute beginner*. Cengage Learning, 3 edition, 2010.

[247] Michael de Plater. Postmortem: Monolith Productions' Middle-earth: Shadow of Mordor. `http://www.gamasutra.com/view/news/234421/Postmortem_Monolith_Productions_Middleearth_Shadow_of_Mordor.php`, Jan 2015.

[248] Fiorella De Rosis, Valeria Carofiglio, Giuseppe Grassano, and Cristiano Castelfranchi. Can computers deliberately deceive? a simulation tool and its application to Turing's imitation game. *Computational Intelligence*, 19(3):235–263, 2003.

[249] Natalie Dehn. Memory in story invention. In *Proc. Cognitive Science Society*, pages 213–215, 1981.

[250] Natalie Dehn. Nlp research at the yale ai project. *ACM SIGART Bulletin*, (79):105–105, 1982.

[251] Natalie Jane Dehn. Computer story-writing: The role of reconstructive and dynamic memory. 1989.

[252] Natlie Dehn. Story generation after Tale-Spin. In *Proc. International Joint Conference on Artificial Intelligence*, volume 81, pages 16–18, 1981.

[253] Christy Dena. *Transmedia practice: Theorising the practice of expressing a fictional world across distinct media and environments*. PhD thesis, University of Sydney Sydney, 2009.

[254] Tim Denee. Bronzemurder. `https://www.timdenee.com/bronzemurder`, 2010. Last accessed: May 11, 2018.

[255] Tim Denee. Oilfurnace. *PC PowerPlay*, 2010.

[256] Tim Denee. Oilfurnace. `https://www.timdenee.com/oilfurnace`, 2010. Last accessed: May 11, 2018.

[257] Daniel C Dennett. The self as a center of narrative gravity. In Frank S Kessel, Pamela M Cole, and Dale L Johnson, editors, *Self and Consciousness: Multiple Perspectives*, pages 275–288. Hillsdale, 1992.

[258] Bella M DePaulo. The many faces of lies. In A.G. Miller, editor, *The Social Psychology of Good and Evil*, pages 303–26. Guilford Press, 2004.

[259] Omkar Deshpande, Serafim Batzoglou, Marcus W Feldman, and L Luca Cavalli-Sforza. A serial founder effect model for human settlement out of africa. *Proc. Royal Society of London B: Biological Sciences*, 276(1655):291–300, 2009.

[260] Tobias Deutsch, Andreas Gruber, Roland Lang, and Rosemarie Velik. Episodic memory for autonomous agents. In *Proc. Human System Interactions*, pages 621–626. IEEE, 2008.

[261] Joshua Lindsay Diaz. Dwarf Fortress gathers at the statue and attends a party. Master's thesis, Massachusetts Institute Of Technology, 2009.

[262] Charles Dickens. *A Christmas Carol and Other Christmas Writings*. Penguin UK, 2003.

[263] Melanie Dickinson, Noah Wardrip-Fruin, and Michael Mateas. Social simulation for social justice. In *Proc. Experimental AI in Games*, 2017.

[264] John M Digman. Personality structure: Emergence of the five-factor model. *Annual Review of Psychology*, 41(1):417–440, 1990.

[265] David E Diller, William Ferguson, Alice M Leung, Brett Benyo, and Dennis Foley. Behavior modeling in commercial games. *Behavior Representation in Modeling and Simulation*, 68, 2004.

[266] Nate DiMeo. The memory palace. `http://thememorypalace.us/`. Last accessed: June 24, 2018.

[267] Edgar Lawrence Doctorow. False documents, 1977.

[268] Steffi Domike, Michael Mateas, and Paul Vanouse. The Recombinant History Apparatus presents: Terminal Time. In Michael Mateas and Phoebe Sengers, editors, *Narrative Intelligence*, pages 155–173. John Benjamins Press, 2003.

[269] Joe Donnelly. Dwarf Fortress drops 'artifact' update with new historical events, prisoner rescues, cover identities. *PC Gamer*, Nov 2017.

[270] Andrew Doull. The death of the level designer: procedural content generation in games. *ASCII Dreams: A Roguelike Developer's Diary*, 2008.

[271] Steven Dow, Blair MacIntyre, and Michael Mateas. Styles of play in immersive and interactive story: case studies from a gallery installation of ar façade. In *Proc. International Conference on Advances in Computer Entertainment Technology*, pages 373–380. ACM, 2008.

[272] Steven Dow, Manish Mehta, Annie Lausier, Blair MacIntyre, and Micheal Mateas. Initial lessons from ar façade, an interactive augmented reality drama. In *Proc. ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. ACM, 2006.

[273] Steven Dow, Manish Mehta, Blair MacIntyre, and Michael Mateas. Ar façade: an augmented reality interactve drama. In *Proc. ACM Symposium on Virtual Reality Software and Technology*, pages 215–216. ACM, 2007.

[274] Jean Dubuffet. Art brut in preference to the cultural arts. *Art & Text*, 27:33, 1988.

[275] Tere Duperrault Fassbender and Richard Logan. *Alone: Orphaned on the Ocean*. Titletown, 2010.

[276] Alessandro Duranti. The audience as co-author: An introduction. *Text-Interdisciplinary Journal for the Study of Discourse*, 6(3):239–248, 1986.

[277] Funda Durupinar, Jan Allbeck, Nuria Pelechano, and Norman Badler. Creating crowd variation with the ocean personality model. In *Proc. International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1217–1220. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[278] Dwarf Fortress Wiki. 40d:personality trait. `http://dwarffortresswiki.org/index.php/40d:Personality_trait`. Last accessed: May 2, 2018.

[279] Dwarf Fortress Wiki. Df2014:legends. `http://dwarffortresswiki.org/index.php/DF2014:Legends`. Last accessed: June 4, 2018.

[280] Dwarf Fortress Wiki. Df2014:personality trait. http://dwarffortresswiki.org/index.php/DF2014:Personality_trait#Facets. Last accessed: June 24, 2018.

[281] Dwarf Fortress Wiki. Df2014:tile. http://dwarffortresswiki.org/index.php/DF2014:Tile. Last accessed: June 26, 2018.

[282] Dwarf Fortress Wiki. Df2014:tissue definition token. http://dwarffortresswiki.org/index.php/DF2014:Tissue_definition_token. Last accessed: May 15, 2018.

[283] Dwarf Fortress Wiki. Df2014:underworld. http://dwarffortresswiki.org/index.php/DF2014:Underworld. Last accessed: May 11, 2018.

[284] Dwarf Fortress Wiki. Save compatibility. http://dwarffortresswiki.org/index.php/Save_compatibility. Last accessed: May 15, 2018.

[285] Dwarf Fortress Wiki. Seed archive. http://dwarffortresswiki.org/index.php/Seed_archive. Last accessed: June 26, 2018.

[286] Dwarf Fortress Wiki. v0.31:personality trait. http://dwarffortresswiki.org/index.php/v0.31:Personality_trait. Last accessed: May 2, 2018.

[287] Dwarf Fortress Wiki. v0.34:personality trait. http://dwarffortresswiki.org/index.php/v0.34:Personality_trait. Last accessed: May 2, 2018.

[288] Michael G Dyer. Artificial intelligence research at the University of California, Los Angeles. *AI Magazine*, 6(3):210, 1985.

[289] Michael George Dyer. In-depth understanding: A computer model of integrated processing for narrative comprehension. Technical report, Yale University, 1982.

[290] Robert Eaglestone. Postmodernism and holocaust denial. 2001.

[291] Earplay. *Codename Cygnus Reactivated*. Earplay, 2018.

[292] Earplay. *Jurassic World Revealed*. Universal Studios Interactive, 2018.

[293] Earplay. *Office Hours*. Future of StoryTelling, 2018.

[294] Earplay. *The Orpheus Device*. White Wolf, 2018.

[295] Earplay. *You and the Beanstalk*. WBUR, 2018.

[296] Vanya Eccles. Procedurally generated background characters. Master's thesis, University of Dublin, Trinity College, 2017.

[297] Douglas Eck. Welcome to Magenta! `https://magenta.tensorflow.org/blog/2016/06/01/welcome-to-magenta`, Jun 2016.

[298] Penelope Eckert. Adolescent social structure and the spread of linguistic change. *Language in Society*, 17(2):183–207, 1988.

[299] Penelope Eckert. *Language variation as social practice: The linguistic construction of identity in Belten High*. Wiley-Blackwell, 2000.

[300] Markus Eger. *Intentional Agents for Doxastic Games*. PhD thesis, North Carolina State University, 2018.

[301] Markus Eger, Camille Barot, and R Michael Young. Impulse: A formal characterization of story. In *OpenAccess Series in Informatics*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[302] Arjan Egges, Sumedha Kshirsagar, and Nadia Magnenat-Thalmann. A model for personality and emotion simulation. In *Proc. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 453–461. Springer, 2003.

[303] Lajos Egri. *The art of dramatic writing: Its basis in the creative interpretation of human motives*. Simon and Schuster, 1960.

[304] Will Eisner. *Comics & sequential art*. Poorhouse Press, 1985.

[305] Mirjam P Eladhari, Anne Sullivan, Gillian Smith, and Josh McCoy. AI-based game design: Enabling new playable experiences. Technical Report UCSC-SOE-11-27, University of California, Santa Cruz, 2011.

[306] Melissa Elliot. Into the deep: it's time to learn how to play Dwarf Fortress. *PCGamer*, Jul 2014.

[307] Clark Elliott and Jacek Brzezinski. Autonomous agents as synthetic characters. *AI Magazine*, 19(2):13, 1998.

[308] Clark Elliott, Jacek Brzezinski, Sanjay Sheth, and Robert Salvatoriello. Story-morphing in the affective reasoning paradigm: generating stories semi-automatically for use with "emotionally intelligent" multimedia agents. In *Proc. International Conference on Autonomous Agents*, pages 181–188. ACM, 1998.

[309] Clark Elliott and Ernst Melchior. Getting to the point: Emotion as a necessary and sufficient element of story construction. In *Working Notes of the AAAI Spring Symposium on Interactive Story Systems*, 1995.

[310] Carolyn Ellis, Tony E Adams, and Arthur P Bochner. Autoethnography: an overview. *Historical Social Research/Historische Sozialforschung*, pages 273–290, 2011.

[311] David K Elson. Detecting story analogies from annotations of time, action and agency. In *Proc. Workshop on Computational Models of Narrative*, 2012.

[312] David K Elson. *Modeling Narra tive Discourse*. PhD thesis, Columbia University, 2012.

[313] David K Elson and Mark O Riedl. A lightweight intelligent virtual cinematography system for machinima production. In *Proc. Artificial Intelligence in Interactive Digital Media*, 2007.

[314] Joshua M Epstein. Agent-based computational models and generative social science. *Complexity*, 4(5):41–60, 1999.

[315] Cumhur Erkut. Call for artworks: The International Conference on Interactive Digital Storytelling. `http://icids2015.aau.dk/exhibition/`, 2015.

[316] Jennifer Edson Escalas. Imagine yourself in the product: Mental simulation, narrative transportation, and persuasion. *Journal of advertising*, 33(2):37–48, 2004.

[317] Markku Eskelinen. Towards computer game studies. *Digital Creativity*, 12(3):175–183, 2001.

[318] Jane Espenson. How to give maris hives, alphabetized. `http://www.janeespenson.com/archives/00000550.php`, Apr 2008. Last accessed: June 25, 2018.

[319] Kellie Michele Evans. Larger than life: Digital creatures in a family of two-dimensional cellular automata. In *Proc. Discrete Mathematics and Theoretical Computer Science*, pages 177–192, 2001.

[320] Max Evans. The many terms in our continuum: 'outsider art'. `http://brutforce.com/many-terms-continuum-outsider-art/`, Jul 2016.

[321] Michael Evans, Tristan Ferne, Zillah Watson, Frank Melchior, Matthew Brooks, Phil Stenton, and Ian Forrester. Creating object-based experiences in the real world. In *Proc. International Broadcasting Convention*. IET, 2016.

[322] Richard Evans. The future of AI in games: A personal view. *Game Developer Magazine*, 2001.

[323] Richard Evans. Re-expressing normative pragmatism in the medium of computation. In *Proc. Collective Intentionality VI*, 2008.

[324] Richard Evans. Modeling individual personalities in The Sims 3. In *Presented at Game Developers' Conference*, 2010.

[325] Richard Evans. Representing personality traits as conditionals. In *Proc. Artificial Intelligence and Simulation of Behavior*, pages 35–42, 2011.

[326] Richard Evans and Emily Short. Versu—a simulationist storytelling system. *Transactions on Computational Intelligence and AI in Games*, 6(2):113–130, 2014.

[327] Richard Prideaux Evans. Computer models of constitutive social practice. In *Fundamental Issues of Artificial Intelligence*, pages 391–411. Springer, 2016.

[328] Exxos. *Captain Blood*. Infogrames, 1988.

[329] Sander Faas. Virtual storyteller. Master's thesis, University of Twente, 2002.

[330] Chris Fairclough. *Story games and the OPIATE system*. PhD thesis, University of Dublin, Trinity College, 2004.

[331] Rino Falcone, Cristiano Castelfranchi, and Fiorella de Rosis. Deceiving in GOLEM: how to strategically pilfer help. In Cristiano Castelfranchi and Yao-Hua Tan, editors, *Trust and Deception in Virtual Societies*, pages 91–109. Springer, 2001.

[332] John Fallon and Mads Haahr. Believable behaviour of background characters in open world games. Master's thesis, University of Dublin, Trinity College, 2013.

[333] Matthew William Fendt and Robert Michael Young. The case for intention revision in stories and its incorporation into IRIS, a story-based planning system. In *Proc. Intelligent Narrative Technologies*, 2011.

[334] Jacques Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley Reading, 1999.

[335] William Ferguson, David Diller, Kerry Moffitt, Bruce Roberts, Will Coon, Michael Mateas, Dan Shapiro, Andrew Stern, Larry LeBron, and Karen Tannenbaum. Immerse: Interactive mentoring for multimodal experiences in realistic social encounters. Technical report, 2015.

[336] Clara Fernández-Vara. Game spaces speak volumes: Indexical storytelling. In *Proc. Digital Games Research Association*. Digital Games Research Association, 2011.

[337] Lucas N Ferreira. Advancement: Generative soundtracks for oral storytelling. `https://www.soe.ucsc.edu/events/advancement-generative-soundtracks-oral-storytelling`, 2018. Last accessed: June 26, 2018.

[338] Ben R Finney. Voyaging canoes and the settlement of polynesia. *Science*, 196(4296):1277–1285, 1977.

[339] Stanley E Fish. How to do things with Austin and Searle: Speech act theory and literary criticism. *MLN*, 91(5):983–1025, 1976.

[340] Adam Fisher. Valley of genius: The uncensored history of Silicon Valley, as told by the hackers, founders, and freaks who made it boom, 2018.

[341] Robert Taylor Fisher. Re: My involvement. `https://groups.google.com/d/msg/rec.arts.int-fiction/LFWFy1ig4Ug/2LtsAFB8F_gJ`, Mar 1991.

[342] Robert Taylor Fisher. Use of utility functions in interactive fiction. `https://groups.google.com/d/msg/rec.arts.int-fiction/UbOh2lE-btw/AwWei4_K_kwJ`, Apr 1992.

[343] Mary Flanagan. *Critical play: radical game design*. MIT Press, 2009.

[344] Richard Fletcher. *Bloodfeud: murder and revenge in Anglo-Saxon England*. Oxford University Press on Demand, 2004.

[345] Catherine Flick, L Meghan Dennis, and Andrew Reinhard. Exploring simulated game worlds: Ethics in the No Man's Sky archaeological survey. *ORBIT*, 1(2), 2017.

[346] Matthieu Fontaine and Shahid Rahman. Fiction, creation and fictionality: An overview. *Methodos. Savoirs et textes*, (10), 2010.

[347] Timothy Foote. Middle-earth genesis. *TIME*, (110):121, Oct 1977.

[348] Kenneth D Forbus and Will Wright. Some notes on programming objects in the sims™. Technical report, Northwestern University, 2001.

[349] Ian Forrester and Anthony Churnside. The creation of a perceptive audio drama, 2012. Presented at the New European Media Summit.

[350] Edward Morgan Forster. *Aspects of the Novel*. Penguin, 1972 [1927].

[351] Michel Foucault. *Archaeology of Knowledge*. Routledge, 1969.

[352] Adam Frank, Andrew Stern, and Ben Resner. Socially intelligent virtual Petz. In *Proc. AAAI Fall Symposium on Socially Intelligent Agents*, number AAAI Technical Report FS-97-02. AAAI Press, 1997.

[353] Ian Frank. Explanations count. In *Proc. AAAI Spring Symposium on Artificial Intelligence and Computer Games*, pages 77–80, 1999.

[354] Priscilla Frank. Should we stop using the term 'outsider art'? *Huffington Post*, Jul 2016.

[355] Gonzalo Frasca. The Sims: Grandmothers are cooler than trolls. *Game Studies*, 1(1), 2001.

[356] Gonzalo Frasca. Ludologists love stories, too: notes from a debate that never took place. In *Proc. Digital Games Research Association*, 2003.

[357] Peter Freebody and Richard C Anderson. Serial position and rated importance in the recall of text. *Discourse processes*, 9(1):31–36, 1986.

[358] Gottlob Frege. Sense and reference. *The Philosophical Review*, 57(3):209–230, 1948.

[359] Scott R French. *Just This Once*. Birch Lane Press, 1993.

[360] Anna Frew and Ian Forrester. Evaluating visual perceptive media. In *Proc. International Conference on Interactive Digital Storytelling*, pages 323–326. Springer, 2017.

[361] Robert W Frick. Interestingness. *British Journal of Psychology*, 83(1):113–128, 1992.

[362] Chris Frith and Uta Frith. Theory of mind. *Current Biology*, 15(17):R644–R645, 2005.

[363] Tracy Fullerton. *Game design workshop: a playcentric approach to creating innovative games*. CRC press, 2008.

[364] Christopher T Funkhouser. IBM poetry: Exploring restriction in computer poems. *Humanities*, 6(1):7, 2017.

[365] Christopher Thompson Funkhouser and Sandy Baldwin. *Prehistoric digital poetry: an archaeology of forms, 1959-1995*. The University of Alabama Press, 2007.

[366] BA Galitsky. The formal scenario and metalanguage support means to reason with it. Technical Report 98, Rutgers University, 1998.

[367] Boris Galitsky. Automatic generation of the multiagent story for the internet advertisement. In *Proc. Symposium on Artificial Intelligence and Creative Language: Stories and Humour*, 1999.

[368] Catherine Gallagher. The rise of fictionality. *The Novel*, 1:336–63, 2006.

[369] Tinsley A Galyean. Guided navigation of virtual environments. In *Proc. ACM Symposium on Interactive 3D Graphics*, pages 103–104. ACM, 1995.

[370] Tinsley A Galyean. Narrative guidance. In *Working Notes of the AAAI Spring Symposium on Interactive Story Systems*, 1995.

[371] Tinsley Azariah Galyean. *Narrative guidance of interactivity*. PhD thesis, Massachusetts Institute of Technology, 1995.

[372] Hello Games. *No Man's Sky*. Sony Interactive Entertainment, 2016.

[373] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object Oriented Software*. Addison-Wesley, 1994.

[374] Jacob Garbe. Delve: A game by Jacob Garbe (project website). `http://project.jacobgarbe.com/`. Last accessed: November 19, 2018.

[375] Jacob Garbe. Games as narrative generators. 2014.

[376] Jacob Garbe. Simulation of history and recursive narrative scaffolding. `http://project.jacobgarbe.com/simulation-of-history-and-recursive-narrative-scaffolding/`, Feb 2018.

[377] Jacob Garbe, A Reed, Melanie Dickinson, M Mateas, and N Wardrip-Fruin. Author assistance visualizations for Ice-Bound, a combinatorial narrative. *Proc. International Conference on the Foundations of Digital Games*, 2014.

[378] Jacob Garbe, Aaron A Reed, Noah Wardrip-Fruin, and Michael Mateas. Ice-Bound: A novel of reconfiguration. In *Proc. International Conference on the Foundations of Digital Games*, 2014.

[379] Maria B Garda. Neo-rogue and the essence of roguelikeness. *Homo Ludens*, 1(5):59–72, 2013.

[380] Martin Gardner. Mathematical games: the fantastic combinations of John Conway's new solitaire game âĂŸlife'. *Scientific American*, (223), 1970.

[381] Michael R Gardner. *An expert writing model for story composition*. PhD thesis, Loughborough University of Technology, 1991.

[382] Oliver Garfield. *Geniacs: Simple Electric Brain Machines and How to Make Them*. Oliver Garfield, Inc., 1955.

[383] Paul L Garvin. Computer participation in linguistic research. *Language*, 38(4):385–389, 1962.

[384] Gérard Genette. *Narrative discourse: An essay in method*. Cornell University Press, 1980 [1972].

[385] Gérard Genette. Fictional narrative, factual narrative. *Poetics Today*, 11(4):755–774, 1990.

[386] Gérard Genette and Jane Lewen. Narrative discourse revisited. 1988 [1983].

[387] Gérard Genette and Marie Maclean. Introduction to the paratext. *New Literary History*, 22(2):261–272, 1991.

[388] Pablo Gervás. Stories from games: Content and focalization selection in narrative composition. In *Actas del Primer Simposio Español de Entretenimiento Digital*, 2013.

[389] Pablo Gervás. Composing narrative discourse for stories of many characters: a case study over a chess game. *Literary and Linguistic Computing*, 29(4):511–531, 2014.

[390] Pablo Gervás. Metrics for desired structural features for narrative renderings of game logs. *Entertainment Computing*, 5(4):245–250, 2014.

[391] Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás. Story plot generation based on CBR. *Knowledge-Based Systems*, 18(4-5):235–242, 2005.

[392] Pablo Gervás and Carlos León. Reading and writing as a creative cycle: the need for a computational model. In *Proc. International Conference on Computational Creativity*, pages 182–191, 2014.

[393] Codruta Liliana Gîrlea, Eyal Amir, and Roxana Girju. Tracking beliefs and intentions in the werewolf game. In *Proc. International Conference on Principles of Knowledge Representation and Reasoning*, pages 638–641, 2014.

[394] Michael Gleicher. Retargetting motion to new characters. In *Proc. Computer Graphics and Interactive Techniques*, pages 33–42. ACM, 1998.

[395] James Gleick. The information: A history, a theory, a flood. 2011.

[396] Piotr J Gmytrasiewicz and Christine L Lisetti. Modeling users' emotions during interactive entertainment sessions. In *Proc. AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, 2000.

[397] Ben Goertzel and Cassio Pennachin. *Artificial general intelligence*, volume 2. Springer, 2007.

[398] Phil Goetz. Notes on using SNePS for interactive fiction. Technical report, Student University of New York, Buffalo, 1993.

[399] Erving Goffman. The presentation of self in everyday life. 1959.

[400] Joseph A Goguen and D Fox Harrell. Information visualisation and semiotic morphisms. In *Studies in Multidisciplinarity*, volume 2, pages 83–97. Elsevier, 2005.

[401] Lewis R Goldberg. Language and individual differences: The search for universals in personality lexicons. *Review of Personality and Social Psychology*, 2(1):141–165, 1981.

[402] Laurence R Goldman. *Child's Play: Myth, Mimesis and Make-Believe*. ERIC, 1998.

[403] Laurence R Goldman and Michael Emmison. Make-believe play among huli children: Performance, myth, and imagination. *Ethnology*, 34(4):225–255, 1995.

[404] Neil M Goldman. *Computer generation of natural language from a deep conceptual base*. PhD thesis, Stanford University, 1974.

[405] Will Goldstone. *Unity game development essentials*. Packt Publishing Ltd., 2009.

[406] Tao Gong and Lan Shuai. Computer simulation as a scientific approach in evolutionary linguistics. *Language Sciences*, 40:12–23, 2013.

[407] Charles Goodwin. Notes on story structure and the organization of participation. *Structures of Social Action*, pages 225–246, 1984.

[408] Andrew Gordon and Reid Swanson. Identifying personal stories in millions of weblog entries. In *Proc. Weblogs and Social Media Data Challenge Workshop*, 2009.

[409] Andrew Gordon and Michael Van Lent. Virtual humans as participants vs. virtual humans as actors. In *Proc. AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, pages 34–38, 2002.

[410] Andrew S Gordon, Qun Cao, and Reid Swanson. Automated story capture from internet weblogs. In *Proc. International Conference on Knowledge Capture*, pages 167–168. ACM, 2007.

[411] Andrew S Gordon and Kavita Ganesan. Automated story capture from conversational speech. In *Proc. International Conference on Knowledge Capture*, pages 145–152. ACM, 2005.

[412] Andrew S Gordon and Reid Swanson. Storyupgrade: Finding stories in internet weblogs. In *Proc. International AAAI Conference on Web and Social Media*, 2008.

[413] Adrian Gradinar, Daniel Burnett, Paul Coulton, Ian Forrester, Matt Watkins, Tom Scutt, and Emma Murphy. Perceptive media–adaptive storytelling for digital broadcast. In *Human–Computer Interaction*, pages 586–589. Springer, 2015.

[414] Stephen Grand, Dave Cliff, and Anil Malhotra. Creatures: Artificial life autonomous software agents for home entertainment. In *Proc. International Conference on Autonomous Agents*, pages 22–29. ACM, 1997.

[415] Jonathan Gratch and Stacy Marsella. Technical details of a domain-independent framework for modeling emotion. Technical report, University of Southern California Information Sciences Institute, 2004.

[416] David Graves. The automatic playwright, 1991.

[417] David Graves. Plot automation. *Journal of Computer Game Design*, 5(1):10–12, 1991.

[418] Donald H Graves. *Writing: Teachers and children at work*. Heinemann, 1983.

[419] Russell D Gray, Alexei J Drummond, and Simon J Greenhill. Language phylogenies reveal expansion pulses and pauses in Pacific settlement. *Science*, 323(5913):479–483, 2009.

[420] Melanie C Green, Timothy C Brock, and Geoff F Kaufman. Understanding media enjoyment: The role of transportation into narrative worlds. *Communication Theory*, 14(4):311–327, 2004.

[421] H Paul Grice. Logic and conversation. *1975*, pages 41–58, 1975.

[422] Joseph E Grimes. La computadora en las investigaciones humanísticas. *Anuario de Letras. Lingüística y Filología*, 5:163–174, 1965.

[423] Joseph E Grimes. Linguistic and anthropological projects using the computer. *The Use of Computers in Anthropology*, pages 515–516, 1965.

[424] Joseph E Grimes, Rogelio E Cardona-Rivera, and James Ryan. Translation of "La Simulación". 2017. Available upon request.

[425] Jason Grinblat. Designing for modularity. In Tanya X Short and Tarn Adams, editors, *Procedural Generation in Game Design*, pages 29–41. AK Peters/CRC Press, 2017.

[426] Jason Grinblat. `https://twitter.com/ptychomancer/status/980968298002006016`, Apr 2018.

[427] Jason Grinblat and C Brian Bucklew. Subverting historical cause & effect: generation of mythic biographies in caves of qud. In *Proc. Procedural Content Generation*, pages 76–82. ACM, 2017.

[428] Jason Grinblat and Charles Brian Bucklew. *Caves of Qud*. Freehold Games, 2010.

[429] Dave Grossman. *Earplay Demo*. Earplay, 2016.

[430] Dave Grossman. *Mr. Robot Daily Five Nine*. 2016.

[431] April Grow, Melanie Dickinson, Johnathan Pagnutti, Noah Wardrip-Fruin, and Michael Mateas. Crafting in games. *Digital Humanities Quarterly*, 11(4), 2017.

[432] Leif Gruenwoldt, Stephen Danton, and Michael Katchabaw. Creating reactive non player character artificial intelligence in modern video games. In *Proc. GameOn North America Conference*, pages 10–17, 2005.

[433] Jozef Gruska. *Quantum computing*. McGraw-Hill London, 1999.

[434] GTA Wiki. Psycho killer. `http://gta.wikia.com/wiki/Psycho_Killer`.

[435] Laxmi Gupta. A variant of Tale-Spin with independent data and rule bases. Master's thesis, Rochester Institute of Technology, 1986.

[436] Gary Gygax and Dave Arneson. *Dungeons and dragons.* Tactical Studies Rules, 1974.

[437] Norma Haan, Roger Millsap, and Elizabeth Hartka. As time goes by: Change and stability in personality over fifty years. *Psychology and aging*, 1(3):220, 1986.

[438] Nicolas Habonneau, Urs Richle, Nicolas Szilas, and Jean E Dumas. 3d simulated interactive drama for teenagers coping with a traumatic brain injury in a parent. In *Proc. International Conference on Interactive Digital Storytelling*, pages 174–182. Springer, 2012.

[439] Steven Hahn. *A nation under our feet: Black political struggles in the rural South from slavery to the Great Migration.* Harvard University Press, 2003.

[440] Mike Hale. The shenanigans behind the slogan. *New York Times*, page C2, Feb 2012.

[441] Alex Haley. *Roots: The Saga of an American Family.* Doubleday, 1976.

[442] Charlie Hall. Dwarf fortress will crush your CPU because creating history is hard. *Polygon*, July 2014.

[443] Robert A Hall. Idiolect and linguistic super-ego. *Studia Linguistica*, 5(1):21–27, 1951.

[444] Morris Halle. Prolegomena to a theory of word formation. *Linguistic Inquiry*, 4(1):3–16, 1973.

[445] Käte Hamburger. The logic of literature. 1973.

[446] Ben Hammersley. *Content syndication with RSS.* O'Reilly Media, Inc., 2003.

[447] Richard J Hand and Mary Traynor. *The radio drama handbook: Audio drama in context and practice.* A&C Black, 2011.

[448] Sarah Harmon and Arnav Jhala. Imaginative recall with story intention graphs. In *Proc. Workshop on Computational Models of Narrative.* Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[449] Sarah Harmon and Arnav Jhala. Revisiting computational models of creative storytelling based on imaginative recall. In *Proc. International Conference on Interactive Digital Storytelling*, pages 170–178. Springer, 2015.

[450] Sarah M Harmon. *Narrative encoding for computational reasoning and adaptation.* PhD thesis, University of California, Santa Cruz, 2017.

[451] D Fox Harrell. Shades of computational evocation and meaning: The GRIOT system and improvisational poetry generation. In *Proc. Digital Arts and Culture*, 2005.

[452] D Fox Harrell. GRIOT's tales of haints and seraphs: A computational narrative generation system. pages 177–182. MIT Press, 2007.

[453] D Fox Harrell. *Phantasmal media: An approach to imagination, computation, and expression.* MIT Press, 2013.

[454] D Fox Harrell, Dominic Kao, Chong-U Lim, Jason Lipshin, and Ainsley Sutherland. The Chimeria platform: User empowerment through expressing social group membership phenomena. In *Proc. Digital Humanities Conference*, 2014.

[455] D Fox Harrell, Dominic Kao, Chong-U Lim, Jason Lipshin, Ainsley Sutherland, Julia Makivic, and Danielle Olson. Authoring conversational narratives in games with the Chimeria platform. In *Proc. International Conference on the Foundations of Digital Games*, 2014.

[456] K David Harrison, Mark Dras, Berk Kapicioglu, et al. Agent-based modeling of the evolution of vowel harmony. In *Proc. North East Linguistic Society*, volume 1, pages 217–236, 2002.

[457] Vi Hart and Nicky Case. *Parable of the Polygons.* 2014.

[458] Charles O Hartman. *Virtual muse: experiments in computer poetry.* Wesleyan University Press, 1996.

[459] Uri Hasson, Ohad Landesman, Barbara Knappmeyer, Ignacio Vallines, Nava Rubin, and David J Heeger. Neurocinematics: The neuroscience of film. *Projections*, 2(1):1–26, 2008.

[460] Andrew F Hayes. Age preferences for same-and opposite-sex partners. *The Journal of Social Psychology*, 135(2):125–133, 1995.

[461] B Hayes-Roth et al. Directed improvisation: A new paradigm for computer games. In *Proc. Computer Game Developers' Conference*, 1995.

[462] Barbara Hayes-Roth. Agents on stage: Advancing the state of the art of AI. In *Proc. International Joint Conference on Artificial Intelligence*, pages 967–971, 1995.

[463] Barbara Hayes-Roth, Lee Brownston, and Robert van Gent. Multiagent collaboration in directed improvisation. In *Proc. International Conference on Multiagent Systems*, pages 148–154, 1995.

[464] Barbara Hayes-Roth and Patrick Doyle. Animate characters. *Autonomous agents and multi-agent systems*, 1(2):195–230, 1998.

[465] Barbara Hayes-Roth, Erik Sincoff, Lee Brownston, Ruth Huard, and Brian Lent. Directed improvisation with animated puppets. In *Conference Companion on Human Factors in Computing Systems*, pages 79–80. ACM, 1995.

[466] Barbara Hayes-Roth, Andrew Stern, and Janet H Murray. Interactive fiction. *IEEE Intelligent Systems and their Applications*, 13(6):12–15, 1998.

[467] Barbara Hayes-Roth and Robert Van Gent. Improvisational puppets, actors, and avatars. In *Proc. Computer Game Developers' Conference*, 1996.

[468] Barbara Hayes-Roth and Robert Van Gent. Story-marking with improvisational puppets. In *Proc. International Conference on Autonomous Agents*, pages 1–7. ACM, 1997.

[469] Robert Hearn. Re: Computer writes a book? `https://groups.google.com/d/msg/comp.ai/9fk1FVwjsTM/HeZBmS56pjYJ`, Jan 1993.

[470] Douglas Heaven. When infinity gets boring: What went wrong with No Man's Sky. *New Scientist*, Sep 2016.

[471] Stephen A Hedges. Dice music in the eighteenth century. *Music & Letters*, 59(2):180–187, 1978.

[472] George E Heidorn. Natural language inputs to a simulation programming system: An introduction. Technical report, Naval Postgraduate School, 1971.

[473] John Heintz. Reference and inference in fiction. *Poetics*, 8(1-2):85–99, 1979.

[474] Satu Heliö. Simulating the storytelling qualities of life: Telling stories with the sims. In *DiGRA Conference*. Citeseer, 2005.

[475] Jessi Hempel. If podcasts are the new blogs, enjoy the golden age while it lasts. *Wired*, Dec 2015.

[476] Jeffrey Henning. A naming language. `http://fiatlingua.org/tag/jeffrey-henning`, 1995. Last accessed: December 14, 2018.

[477] Sergio Poo Hernandez, Vadim Bulitko, and Marcia Spetch. Keeping the player on an emotional trajectory in interactive storytelling. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, 2015.

[478] Suzanne Hidi. Interest and its contribution as a mental resource for learning. *Review of Educational Research*, 60(4):549–571, 1990.

[479] Suzanne Hidi and Valerie Anderson. Situational interest and its impact on reading and expository writing. *The role of interest in learning and development*, 11:213–214, 1992.

[480] Suzanne Hidi and William Baird. Interestingness—a neglected variable in discourse processing. *Cognitive Science*, 10(2):179–194, 1986.

[481] Suzanne E Hidi and John A McLaren. Motivational factors and writing: The role of topic interestingness. *European Journal of Psychology of Education*, 6(2):187–197, 1991.

[482] James A Hijiya. American gravestones and attitudes toward death: A brief history. *Proceedings of the American Philosophical Society*, 127(5):339–363, 1983.

[483] Wycliffe Aber Hill. *Ten million photoplay plots*. The Feature Photodrama Company, 1919.

[484] Wycliffe Aber Hill. *Index book for use with the Plot Genie: Action Adventure*. Ernest E. Gagnon Company, 1931.

[485] Wycliffe Aber Hill. *Index book for use with the Plot Genie: Romance without melodrama*. Ernest E. Gagnon Company, 1931.

[486] Wycliffe Aber Hill. *The Plot Genie: Index General Formula*. Ernest E. Gagnon Company, 1931.

[487] Wycliffe Aber Hill. *Encyclopedia of Comedy with Index for use with the Plot Genie*. Ernest E. Gagnon Company, 1936.

[488] Wan Ching Ho. *Computational memory architectures for autobiographic and narrative virtual agents*. PhD thesis, University of Hertfordshire, 2005.

[489] Wan Ching Ho, Kerstin Dautenhahn, and Chrystopher L Nehaniv. Comparing different control architectures for autobiographic agents in static virtual environments. In *Proc. Intelligent Virtual Agents*, pages 182–191. Springer, 2003.

[490] Wan Ching Ho, Kerstin Dautenhahn, and Chrystopher L Nehaniv. Autobiographic agents in dynamic virtual environments-performance comparison for different memory control architectures. In *Proc. IEEE Congress on Evolutionary Computation*, pages 573–580. IEEE, 2005.

[491] Wan Ching Ho, Kerstin Dautenhahn, and Chrystopher L Nehaniv. A study of episodic memory-based learning and narrative structure for autobiographic agents. In *Proc. Adaptation in Artificial and Biological Systems*, pages 26–29, 2006.

[492] Wan Ching Ho, Kerstin Dautenhahn, and Chrystopher L Nehaniv. Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model. *Connection Science*, 20(1):21–65, 2008.

[493] Wan Ching Ho, Kerstin Dautenhahn, Chrystopher L Nehaniv, and René Te Boekhorst. Sharing memories: An experimental investigation with multiple autonomous autobiographic agents. In *Intelligent Autonomous Systems*. IOS Press, 2004.

[494] Rania A Hodhod and Brian S Magerko. Computational creativity: Improv agents and conceptual blending. *International Journal of Cognitive Informatics and Natural Intelligence*, 8(2):1–14, 2014.

[495] Marissa Hoek, Mariët Theune, and Jeroen Linssen. Generating game narratives with focalization and flashbacks. In *Proc. Games and Natural Language Processing*, pages 25–55, 2014.

[496] Douglas R Hofstadter. *Gödel, Escher, Bach*. Vintage Books, 1980.

[497] Stuart Hogarth, Gail Javitt, and David Melzer. The current landscape for direct-to-consumer genetic testing: legal, ethical, and policy issues. *Annual Review of Genomics and Human Genetics*, 9:161–182, 2008.

[498] August B Hollingshead. Four factor index of social status. 1975.

[499] John Hollowell. *Fact and fiction: The new journalism and the nonfiction novel*. University of North Carolina Press, 1977.

[500] Sam Holmes. Gay marriage: The database engineering perspective. `https://web.archive.org/web/20100221005910/https://qntm.org/gay`, Nov 2008.

[501] Keith J Holyoak and Paul Thagard. Analogical mapping by constraint satisfaction. *Cognitive Science*, 13(3):295–355, 1989.

[502] Jonathan David Hook. Facts, interactivity and videotape: Exploring the design space of data in interactive video storytelling. In *Proc. ACM TVX International Conference on Interactive Experiences for Television and Online Video*. ACM, 2018.

[503] Ian Horswill. Polly: A vision-based artificial agent. In *Proc. AAAI*, pages 824–829, 1993.

[504] Ian Horswill. Punch and Judy AI playset: A generative farce manifesto or: The tragical comedy or comical tragedy of predicate calculus. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, pages 14–19, 2012.

[505] Ian Horswill. Science considered harmful. In *Proc. Intelligent Narrative Technologies*, pages 82–85, 2013.

[506] Ian Horswill. Game design for classical AI. In *Proc. Experimental AI in Games*, pages 28–34, 2014.

[507] Ian Horswill. MKULTRA. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, pages 223–225, 2015.

[508] Ian Horswill. Sonification of character reasoning. *Artificial and Computational Intelligence in Games: Integration*, page 234, 2015.

[509] Ian D Horswill. Architectural issues for compositional dialog in games. In *Proc. Games and Natural Language Processing*, 2014.

[510] Ian D Horswill. Fiascomatic: A framework for automated Fiasco playsets. In *Proc. Experimental AI in Games*, pages 22–28, 2015.

[511] Ian D Horswill. Dear LeaderâĂŹs Happy Story Time: A party game based on automated story generation. In *Proc. Experimental AI in Games*, pages 39–45. AAAI, 2016.

[512] Ian Douglas Horswill. *Specialization of perceptual processes.* PhD thesis, Massachusetts Institute of Technology, 1993.

[513] Eduard Hovy. Generating natural language under pragmatic constraints. *Journal of Pragmatics*, 11(6):689–719, 1987.

[514] Eduard Hovy. *Generating natural language under pragmatic constraints.* PhD thesis, Yale University, 1987.

[515] Zhichao Hu, Michelle Dick, Chung-Ning Chang, Kevin Bowden, Michael Neff, Jean E Fox Tree, and Marilyn A Walker. A corpus of gesture-annotated dialogues for monologue-to-dialogue generation from personal narratives. In *Proc. Language Resources*, 2016.

[516] Andrew Hugill. The origins of electronic music. *The Cambridge Companion to Electronic Music*, pages 7–23, 2007.

[517] Eric A. Hulteen. Media room description, 1983.

[518] Lloyd Humberstone. *The connectives*. MIT Press, 2011.

[519] David Hume. *A treatise of human nature*. Courier Corporation, 2003.

[520] Linda Hutcheon et al. *A poetics of postmodernism: history, theory, fiction*. Routledge, 1988.

[521] Matti Hyvärinen. Revisiting the narrative turns. *Life Writing*, 7(1):69–82, 2010.

[522] Immortal-D. The hall of legends, 5th ed. revised. `http://www.bay12forums.com/smf/index.php?topic=168375.0`, Nov 2017. Last accessed: June 27, 2018.

[523] Bas in het Veld, Ben Kybartas, Rafael Bidarra, and John-Jules Ch Meyer. Procedural generation of populations for storytelling. In *Proc. Procedural Content Generation in Games*. 2015.

[524] Peter Ingebretson and Max Rebuschatis. Concurrent interactions in The Sims 4. `http://twvideo01.ubm-us.net/o1/vault/GDC2014/Presentations/Peter_Ingebretson_Concurrent_Interactions_In.pdf`, Apr 2018.

[525] Reflections Interactive. *Driver: You Are the Wheelman*. GT Interactive Software, 1999.

[526] Asghar Iran-Nejad. Cognitive and affective causes of interest and liking. *Journal of Educational Psychology*, 79(2):120, 1987.

[527] Katherine Isbister. *How Games Move Us: Emotion by Design*. MIT Press, 2016.

[528] Wolfgang Iser. The reality of fiction: a functionalist approach to literature. *New Literary History*, 7(1):7–38, 1975.

[529] Damián Isla. Third Eye Crime: Building a stealth game around occupancy maps. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2013.

[530] Damián Isla and Bruce Blumberg. New challenges for character-based AI for games. In *Proc. AAAI Spring Symposium on AI and Interactive Entertainment*, 2002.

[531] Mizuko Ito. Virtually embodied: The reality of fantasy in a multi-user dungeon. *Internet Culture*, pages 87–109, 1997.

[532] John L Jacko. Generating narrative in an interactive fiction game. Master's thesis, West Virginia University, 2018.

[533] Mikhail Jacob and Brian Magerko. Creative arcs in improvised human-computer embodied performances. In *Proc. International Conference on the Foundations of Digital Games*, page 62. ACM, 2018.

[534] Mikhail Jacob, Alexander Zook, and Brian Magerko. Viewpoints ai: Procedurally representing and reasoning about gestures. In *Proc. Digital Games Research Association*, 2013.

[535] Dale Jacquette. *Meinongian logic: The semantics of existence and nonexistence*, volume 11. Walter de Gruyter, 2011.

[536] Marie Jamora and Jason McLagan. Slamdance DIG Spotlight—Bad News. `https://www.youtube.com/watch?v=jrrpLFS9Zcw`, May 2017.

[537] Dusan Jan. *Virtual extras: conversational behavior simulation for background virtual humans.* PhD thesis, University of Southern California, 2012.

[538] Kerry L Jang, W John Livesley, and Philip A Vemon. Heritability of the big five personality dimensions and their facets: a twin study. *Journal of personality*, 64(3):577–592, 1996.

[539] Fotis Jannidis. Event-sequences, plots and narration in computer games. *Dichtung Digital*, pages 281–305, 2007.

[540] A Jaya and GV Uma. A novel approach for intelligent reasoning of stories using ontology. *International Journal of Futuristic Computer Application*, 2010.

[541] Henry Jenkins. Game design as narrative architecture. pages 118–130. MIT Press, 2004.

[542] Henry Jenkins. Transmedia storytelling. *Volume*, (1):56, 2009.

[543] Lane Jennings. Word-art for people: Toward a future of user-friendly poetry and interactive fiction. *Communication Quarterly*, 32(2):113–119, 1984.

[544] Marc G Jensen. John Cage, chance operations, and the chaos game: Cage and the "i ching". *The Musical Times*, 150(1907):97–102, 2009.

[545] Arnav Jhala. Discourse and camera control in interactive narratives. *Handbook of Digital Games and Entertainment Technologies*, pages 349–359, 2017.

[546] Arnav Jhala and R Michael Young. Intelligent machinima generation for visual storytelling. In *Artificial Intelligence for Computer Games*, pages 151–170. Springer, 2011.

[547] Oliver P John and Sanjay Srivastava. The big five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of Personality: Theory and Research*, 2(1999):102–138, 1999.

[548] Mark R Johnson. Modelling cultural, religious and political affiliation in artificial intelligence decision-making. In *Proc. AI & Games Symposium*. 2015.

[549] Michael Boyle Johnson. *WAVESworld, a testbed for constructing 3D semi-autonomous animated characters*. PhD thesis, Massachusetts Institute of Technology, 1995.

[550] Keith Johnstone. *Impro: Improvisation and the theatre*. Routledge, 2012.

[551] Chuck Jones. *Chuck Amuck: The life and times of an animated cartoonist*. Farrar, Straus & Giroux, 1989.

[552] Kelly Lee Jones. *Wonks, marathoners, and storytellers: Describing the participatory culture of NaNoWriMo*. PhD thesis, Mercer University, 2013.

[553] Anna Jordanous. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, 4(3):246–279, 2012.

[554] Paul Lincoln Juell. *Improvements in the style of computer generated natural language text*. PhD thesis, The Ohio State University, 1981.

[555] Jesper Juul. Games telling stories. *Game studies*, 1(1), 2001.

[556] Jesper Juul. The open and the closed: Games of emergence and games of progression. In Frans Mäyrä, editor, *Proc. of Computer Games and Digital Cultures Conference*. Tampere University Press, 2002.

[557] Jesper Juul. *Half-Real: Video Games Between Real Rules and Fictional Worlds*. MIT Press, 2011.

[558] Ed Kahn and Sheldon Klein. Automatic novel writing—an exchange. *SIGART Newsletter*, pages 3–5, Apr 1974.

[559] Edward Paul Kahn. *Foundations of a computable theory of narrative*. PhD thesis, University of California, Berkeley, 1973.

[560] Subhash C Kak. The paninian approach to natural language processing. *International Journal of Approximate Reasoning*, 1(1):117–130, 1987.

[561] Michiko Kakutani. Bending the truth in a million little ways. *New York Times*, Jan 2006.

[562] Eric Kaltman. *The Stabilization, Exploration, and Expression of Computer Game History*. PhD thesis, University of California, Santa Cruz, 2017.

[563] Mark Kantrowitz. GLINDA: Natural language text generation in the Oz interactive fiction project. Technical Report CMU-CS-90-158, 1990.

[564] Mark Kantrowitz. Re: Grammar/style reformatter. `https://groups.google.com/d/msg/sci.lang/cu3t3jfF4sc/ThI-r1ceGO0J`, Feb 1992.

[565] Mark Kantrowitz. Bibliography of research in natural language generation. Technical report, Carnegie Mellon University, 1993.

[566] Mark Kantrowitz and Joseph Bates. Integrated natural language generation systems. In *Aspects of Automated Natural Language Generation*, pages 13–28. Springer, 1992.

[567] Veli-Matti Karhulahti. Do videogames simulate? virtuality and imitation in the philosophy of simulation. *Simulation & Gaming*, 46(6):838–856, 2015.

[568] Stephen Karpinskyj, Fabio Zambetta, and Lawrence Cavedon. Video game personalisation techniques: A comprehensive survey. *Entertainment Computing*, 5(4):211–218, 2014.

[569] Isaac Karth. Preliminary poetics of procedural generation in games. In *Proc. Digital Games Research Association*, 2018.

[570] Deborah A Kashy and Bella M DePaulo. Who lies? *Journal of Personality and Social Psychology*, 70(5):1037, 1996.

[571] Sumitra Mangesh Katre et al. *Aáźčáźŋādhyāyī of Pāṇini*. Motilal Banarsidass, 1989.

[572] Martin Kay. Chart generation. In *Proc. Association for Computational Linguistics*, pages 200–204. Association for Computational Linguistics, 1996.

[573] Darius Kazemi and Hugo van Kemenade. Nanogenmo. `https://nanogenmo.github.io`. Last accessed: June 5, 2018.

[574] Garrison Keillor. *Lake Wobegon Days*. Penguin, 1990.

[575] Margaret Thomas Kelso, Peter Weyhrauch, and Joseph Bates. Dramatic presence. *PRESENCE: Teleoperators & Virtual Environments*, 2(1):1–15, 1993.

[576] George B Kerferd. Gorgias on nature or that which is not. *Phronesis*, pages 3–25, 1955.

[577] Sarah Kessler. 'truman show'-style online game uses actors, not animation. `https://mashable.com/2012/04/24/alternet-reality`, Apr 2012.

[578] Jaegwon Kim. Multiple realization and the metaphysics of reduction. *Philosophy and Phenomenological Research*, 52(1):1–26, 1992.

[579] Juyun Kim and Stephen Petrina. Artificial life rights: Facing moral dilemmas through The Sims. *Educational Insights*, 10(2), 2006.

[580] Mijin Kim and Young Yim Doh. Computational modeling of players' emotional response patterns to the story events of video games. *IEEE Transactions on Affective Computing*, 8(2):216–227, 2017.

[581] Sung-il Kim. Causal bridging inference: A cause of story interestingness. *British Journal of Psychology*, 90(1):57–71, 1999.

[582] Walter Kintsch. Learning from text, levels of comprehension, or: Why anyone would read a story anyway. *Poetics*, 9(1-3):87–98, 1980.

[583] Stephen Cole Kleene. Representation of events in nerve nets and finite automata. Technical Report RM-704, The Rand Corporation, Dec 1951.

[584] W Bastiaan Kleijn and Kuldip K Paliwal. *Speech Coding and Synthesis*. Elsevier Science Inc., 1995.

[585] Sheldon Klein. Dynamic simulation of historical change in language using Monte Carlo techniques. Technical report, System Development Corporation, 1964.

[586] Sheldon Klein. Automatic paraphrasing in essay format. *Mechanical Translation & Computational Linguistics*, 8(3-4):68–83, 1965.

[587] Sheldon Klein. Control of style with a generative grammar. *Language*, 41(4):619–631, 1965.

[588] Sheldon Klein. Some components of a program for dynamic modelling of historical change in language. In *Proc. Conference on Computational linguistics*, pages 1–17. Association for Computational Linguistics, 1965.

[589] Sheldon Klein. Historical change in language using Monte Carlo techniques. *Mechanical Translation & Computational Linguistics*, 9(3-4):67–82, 1966.

[590] Sheldon Klein. Automatic novel writing. In *Current Research and Development in Scientific Documentation*, number 15, pages 417–418. National Science Foundation, 1969.

[591] Sheldon Klein. Computer simulation of language contact models. In Roger W Shuy and Charles-James N Bailey, editors, *Towards Tomorrow's Linguistics*, pages 276–290. Georgetown University Press, 1974.

[592] Sheldon Klein. Meta-compiling text grammars as a model for human behavior. In *Proc. Theoretical Issues in Natural Language Processing*, pages 84–88. Association for Computational Linguistics, 1975.

[593] Sheldon Klein. The analogical foundations of creativity in language, culture & the arts:" the upper paleolithic to 2100ce. In *Language, Vision, and Music: Selected Papers from the 8th International Workshop on the Cognitive Science of Natural Language Processing*, volume 35, page 347. John Benjamins Publishing, 2002.

[594] Sheldon Klein. The invention of computationally plausible knowledge systems in the Upper Palaeolithic. *The Origins of Human Behaviour*, page 67, 2003.

[595] Sheldon Klein, John F Aeschlimann, Matthew A Appelbaum, David F Blasiger, Elizabeth J Curtis, Mark Foster, S David Kalish, Scott J Kamin, Ying-Da Lee, Lynne A Price, and David F Salsieder. Modelling propp and lévi-strauss in meta-symbolic simulation system. computer sciences department, university of wisconsin. Technical Report TR226, University of Wisconsin, 1974.

[596] Sheldon Klein, John F Aeschlimann, Matthew A Appelbaum, David F Blasiger, Elizabeth J Curtis, Mark Foster, S David Kalish, Scott J Kamin, Ying-Da Lee, Lynne A Price, and David F Salsieder. Modeling propp and lévi-strauss in a metasymbolic simulation system. *Patterns in Oral Literature*, pages 141–222, 1977.

[597] Sheldon Klein, John F Aeschlimann, David F Balsiger, Steven L Converse, Claudine Court, Mark Foster, Robin Lao, John D Oakley, and Joel Smith. Automatic novel writing: A status report. Technical Report TR186, University of Wisconsin, 1973.

[598] Sheldon Klein, JL Bradshaw, Stevan Harnad, David YF Ho, Bruce Holbrook, J Anthony Paredes, Robert A Rubinstein, Warren D TenHouten,

and John A Young. Analogy and mysticism and the structure of culture [and comments and reply]. *Current Anthropology*, 24(2):151–180, 1983.

[599] Sheldon Klein, Michael A Kuppin, and Kirby A Meives. Monte Carlo simulation of language change in Tikopia & Maori. In *Proc. Computational Linguistics*, pages 1–27. Association for Computational Linguistics, 1969.

[600] Sheldon Klein and James R Meehan. Ai forum. *SIGART Newsletter*, 61:2–4, Feb 1977.

[601] Sheldon Klein, John D Oakley, David J Suurballe, and Robert A Ziesemer. A program for generating reports on the status and history of stochastically modifiable semantic models of arbitrary universes. Technical Report TR142, University of Wisconsin, 1971.

[602] Sheldon Klein, David A Ross, Mark S Manasse, Johanna Danos, Mark S Bickford, Kendall I Jensen, Walter A Burt, Glenn D Blank, and Walter T Blanks. Propositional & analogical generation of coordinated verbal, visual & musical texts. *ACM SIGART Bulletin*, (79):104–104, 1982.

[603] Sheldon Klein and Robert F Simmons. A computational approach to grammatical coding of English words. *Journal of the Association for Computing Machinery*, 10(3):334–347, 1963.

[604] Sheldon Klein and Robert F Simmons. Syntactic dependence and the computer generation of coherent discourse. *Mechanical Translation*, 7(2):50–61, 1963.

[605] Alison Knowles and James Tenney. *A House of Dust*. König, 1970.

[606] Donald E Knuth. Semantics of context-free languages. *Mathematical Systems Theory*, 2(2):127–145, 1968.

[607] Donald E Knuth. The genesis of attribute grammars. In *Attribute Grammars and Their Applications*, pages 1–12. Springer, 1990.

[608] Manfred Kochen. On the generality of PARRY, Colby's paranoia model. *Behavioral and Brain Sciences*, 4(4):540–541, 1981.

[609] Rebecca Koenig. A new department marks the rise of a discipline: 'computational media'. *The Chronicle of Higher Education*, Oct 2014.

[610] Hartmut Koenitz. Five theses for interactive digital narrative. In *Proc. International Conference on Interactive Digital Storytelling*, pages 134–139. Springer, 2014.

[611] Hartmut Koenitz. Interactive storytelling paradigms and representations: a humanities-based perspective. In Ryohei Nakatsu, Matthias Rauterberg, and Paolo Ciancarini, editors, *Handbook of Digital Games and Entertainment Technologies*, pages 361–375. Springer, 2017.

[612] Janet Kolodner. *Case-based reasoning*. Morgan Kaufmann, 1993.

[613] Janet L Kolodner. Reconstructive memory: A computer model. *Cognitive Science*, 7(4):281–328, 1983.

[614] Richard E Korf. Planning as search: A quantitative approach. *Artificial Intelligence*, 33(1):65–88, 1987.

[615] A Krapp. Interest, learning and academic achievement. In *Proc. Third European Conference of Learning and Instruction*, 1989.

[616] Martin Kreiswirth. Narrative turn in the humanities. In David Herman, Manfred Jahn, and Marie-Laure Ryan, editors, *Routledge Encyclopedia of Narrative Theory*, pages 377–382. Taylor & Francis, 2005.

[617] Michael Kriegel and Ruth Aylett. Emergent narrative as a novel framework for massively collaborative authoring. In *Proc. International Workshop on Intelligent Virtual Agents*, pages 73–80. Springer, 2008.

[618] Michael Kriegel, Ruth Aylett, Joao Dias, and Ana Paiva. An authoring tool for an emergent narrative storytelling system. In *Proc. AAAI Fall Symposium on Intelligent Narrative Technologies*, pages 55–62, 2007.

[619] Saul A Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[620] JC Kroon. Nemesis narratives: The relationship between embedded and emergent narrative in Middle Earth: Shadow of Mordor. Master's thesis, Utrecht University, 2016.

[621] Thomas S Kuhn. *The structure of scientific revolutions*. University of Chicago Press, 1962.

[622] Karin Kukkonen. Bayesian narrative: Probability, plot and the shape of the fictional world. *Anglia*, 132(4):720–739, 2014.

[623] Annakaisa Kultima, Janne Paavilainen, Jaakko Stenros, and Frans Mäyrä. From social play to social games and back: The emergence and development of social network games. In *New Perspectives on the Social Aspects of Digital Gaming*, pages 11–32. Routledge, 2017.

[624] Cameron Kunzelman. The nonhuman lives of videogames. Master's thesis, Georgia State University, 2014.

[625] Ben Kybartas and Rafael Bidarra. A survey on story generation techniques for authoring computational narratives. *Transactions on Computational Intelligence and AI in Games*, 9(3):239–253, 2017.

[626] Ben Kybartas, Clark Verbrugge, and Jonathan Lessard. Subject and subjectivity: A conversational game using possible worlds. In *Proc. International Conference on Interactive Digital Storytelling*, pages 332–335. Springer, 2017.

[627] William Labov. The social motivation of a sound change. *Word*, 19(3):273–309, 1963.

[628] William Labov. On the mechanism of linguistic change. *Georgetown Monographs on Language and Linguistics*, 18:91–114, 1965.

[629] William Labov. *Language in the inner city: Studies in the Black English vernacular*, volume 3. University of Pennsylvania Press, 1972.

[630] William Labov. The intersection of sex and social class in the course of linguistic change. *Language variation and change*, 2(2):205–254, 1990.

[631] William Labov. Some further steps in narrative analysis. *Journal of Narrative and Life History*, 7(1):395–415, 1997.

[632] William Labov. Narrative pre-construction. *Narrative inquiry*, 16(1):37–45, 2006.

[633] William Labov. Transmission and diffusion. *Language*, 83(2):344–387, 2007.

[634] William Labov. Where should i begin. In *Telling Stories: Language, Narrative, and Social Life*, pages 7–22. Georgetown University Press, 2010.

[635] William Labov, Paul Cohen, and Clarence Robins. A study of the non-standard english of Negro and Puerto Rican speakers in New York City. volume II: The use of language in the speech community. Technical Report Cooperative Research Project No. 3288, U.S. Department of Health, Education, & Welfare Office of Education, 1968.

[636] William Labov and David Fanshel. *Therapeutic discourse: Psychotherapy as conversation.* Academic Press, 1977.

[637] William Labov and Joshua Waletzky. Narrative analysis: Oral versions of personal experience. *Journal of Narrative and Life History*, 7(1):3–38, 1997.

[638] Ville Matias Lampi. Looking behind the text-to-be-seen: Analysing Twitter bots as electronic literature. Master's thesis, Aalto University School of Arts, Design and Architecture, 2017.

[639] Irene Langkilde and Kevin Knight. Generation that exploits corpus-based statistical knowledge. In *Proc. Association for Computational Linguistics*, pages 704–710. Association for Computational Linguistics, 1998.

[640] Christopher G Langton. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22(1–3):120–149, 1986.

[641] Christopher G Langton. *Artificial life: An overview.* MIT Press, 1997.

[642] Kaisu Lankinen, Eero Smeds, Pia Tikka, Elina Pihko, Riitta Hari, and Miika Koskinen. Haptic contents of a movie dynamically engage the spectator's sensorimotor cortex. *Human brain mapping*, 37(11):4061–4068, 2016.

[643] Petri Lankoski. Models for story consistency and interestingness in single-player RPGs. In *Proc. International Conference on Making Sense of Converging Media*, page 246. ACM, 2013.

[644] François Lareau, Mark Dras, and Robert Dale. Detecting interesting event sequences for sports reporting. In *Proc. European Workshop on Natural Language Generation*, pages 200–205. Association for Computational Linguistics, 2011.

[645] CB Larsen and BC Petersen. Interactive storytelling in a multimodal environment. Technical report, Aalborg University, 1999.

[646] Charles U Larson. The evoked rhetorical vision of Lake Wobegon: A textual analysis of "A Prairie Home Companion". *Journal of Radio Studies*, 1(1-2):55–69, 1992.

[647] Charles U Larson and Christine Oravec. A Prairie Home Companion and the fabrication of community. *Critical Studies in Media Communication*, 4(3):221–244, 1987.

[648] Sarah Larson. "The Memory Palace": History in escapist vignettes. *The New Yorker*, Jan 2018.

[649] John Lasseter. Principles of traditional animation applied to 3D computer animation. In *Proc. ACM SIGGRAPH*, volume 21, pages 35–44. ACM, 1987.

[650] Brenda Laurel. First-person simulations as learning environments, 1982.

[651] Brenda Laurel. Interactivity: Conversation with Michael Naimark, 1983.

[652] Brenda Laurel. Notes on interestingness in interactive stories: Some playwriting principles, 1983.

[653] Brenda Laurel. Toward the design of an interactive fantasy system: Description and functional requirements, 1983.

[654] Brenda Laurel. *Toward the design of a computer-based interactive fantasy system*. PhD thesis, The Ohio State University, 1986.

[655] Brenda Laurel. Atari research memos on on the subject of interactive fantasy and related topics, 1989.

[656] Brenda Laurel. *Computers as theatre*. Addison-Wesley, 1991.

[657] Brenda Laurel and Eric A. Hulteen. Media room event: Revised proposal, 1983.

[658] Lucian Leahu, Phoebe Sengers, and Michael Mateas. Interactionist AI and the promise of ubicomp, or, how to put your box in the world without putting the world in your box. In *Proc. International Conference on Ubiquitous Computing*, pages 134–143. ACM, 2008.

[659] Marc LeBlanc. Formal design tools: Feedback systems and the dramatic structure of competition. `http://algorithmancy.8kindsoffun.com/cgdc99.ppt`, 1999.

[660] Marc LeBlanc. Formal design tools: Emergent complexity, emergent narrative. `http://algorithmancy.8kindsoffun.com/gdc2000.ppt`, 2000.

[661] Alisa Lebow. Faking what? making a mockery of documentary. In Alexandra Juhasz and Jesse Lerner, editors, *F is for Phony: Fake Documentary and Truth's Undoing*, pages 223–237. 2006.

[662] Michael Lebowitz. *Generalization and Memory in an Integrated Understanding System*. PhD thesis, Yale University, 1980.

[663] Michael Lebowitz. Creating a story-telling universe. In *Proc. International Joint Conference on Artificial Intelligence*, pages 63–65, 1983.

[664] Michael Lebowitz. Memory-based parsing. *Artificial Intelligence*, 21(4):363–404, 1983.

[665] Michael Lebowitz. Creating characters in a story-telling universe. *Poetics*, 13(3):171–194, 1984.

[666] Michael Lebowitz. Story-telling as planning and learning. *Poetics*, 14(6):483–502, 1985.

[667] Michael Lebowitz. Planning stories. In *Proc. Cognitive Science Society*, pages 234–242, 1987.

[668] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[669] Greg Lee, Vadim Bulitko, and Elliot A Ludvig. Automated story selection for color commentary in sports. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(2):144–155, 2014.

[670] Gregory MK Lee. *Automated story-based commentary for sports*. PhD thesis, University of Alberta, 2012.

[671] Harper Lee. To kill a mockingbird. 1960.

[672] Mark G Lee. A model of story generation. Master's thesis, University of Manchester, 1994.

[673] Young-Seol Lee and Sung-Bae Cho. Exploiting mobile contexts for Petri-net to generate a story in cartoons. *Applied Intelligence*, 34(1):1–18, 2011.

[674] Douglas B Lenat. *AM: An artificial intelligence approach to discovery in mathematics as heuristic search*. PhD thesis, 1976.

[675] Douglas B Lenat and John Seely Brown. Why AM and EURISKO appear to work. *Artificial Intelligence*, 23(3):269–294, 1984.

[676] Eric H Lenneberg. *The biological foundations of language*. John Wiley and Sons, 1967.

[677] Tom Leonard. Building an AI sensory system: Examining the design of Thief: The Dark Project. In *Game Development Conference*, 2003.

[678] Jonathan Lessard. Fahrenheit and the premature burial of interactive movies. *Eludamos: Journal for Computer Game Culture*, 3(2):195–205, 2009.

[679] Jonathan Lessard. Design rationale for natural language game conversations. In *Proc. International Conference on the Foundations of Digital Games*, 2015.

[680] Jonathan Lessard. Designing natural-language game conversations. In *Proc. International Joint Conference of DiGRA and FDG*, 2016.

[681] Jonathan Lessard and Dominic Arsenault. The character as subjective interface. In *Proc. International Conference on Interactive Digital Storytelling*, pages 317–324. Springer, 2016.

[682] Jonathan Lessard, Etienne Brunelle-Leclerc, Timothy Gottschalk, Marc-Antoine Jetté-Léger, Odile Prouveur, and Christopher Tan. Striving for author-friendly procedural dialogue generation. In *Proc. Non-Player Characters and Social Believability in Games*. ACM, 2017.

[683] Jonathan Lessard, Marc-Antoine Jetté-Léger, Alexandra Bachmayer, Etienne Brunelle-Leclerc, Christopher Venner-Tan, Odile Prouveur, and Timothy Gottschalk. *Hammurabi*. LabLabLab, 2017.

[684] Claude Lévi-Strauss. L'analyse morphologique des contes russes. *International Journal of Slavic Linguistics and Poetics*, 3:122–149, 1960.

[685] D Levy and T Marsland. The icca best annotation award for 1995. *ICGA Journal*, 19(2):135–136, 1996.

[686] David Lewis. Truth in fiction. *American Philosophical Quarterly*, 15(1):37–46, 1978.

[687] Boyang Li and Mark O Riedl. An offline planning approach to game plotline adaptation. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2010.

[688] Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. Designer modeling for Sentient Sketchbook. In *Proc. Computational Intelligence and Games*, pages 1–8. IEEE, 2014.

[689] Bernhard Lindemann. Text as process: an integrated view of a science of texts. *Journal of Literary Semantics*, 12(1):5–41, 1983.

[690] Vachel Lindsay. *The Art of the Motion Picture*. The Macmillan Company, 1916.

[691] Andro Linklater. *Measuring America: How the United States was shaped by the greatest land sale in history*. Findaway World LLC, 2009.

[692] Deborah E Lipstadt. *Denying the Holocaust: The growing assault on truth and memory*. Simon and Schuster, 1993.

[693] Hugo Liu and Push Singh. Makebelieve: Using commonsense knowledge to generate stories. In *Proc. Innovative Applications Of Artificial Intelligence*, pages 957–958, 2002.

[694] Daniel Livingstone, Sandy Louchart, and Stuart Jeffrey. Archaeological storytelling in games. In *Proc. Workshop on Playing With History: Games, Antiquity and History*, 2016.

[695] JC Loehlin. The personality of aldous. *Discovery*, 23(7):23–26, 1962.

[696] John C Loehlin. A computer program that simulates personality. *Computer Simulation of Personality*, pages 189–211, 1963.

[697] John C Loehlin. " interpersonal" experiments with a computer model of personality. *Journal of personality and social psychology*, 2(4):580, 1965.

[698] John C Loehlin. Machines with personality. *Science Journal*, 1969.

[699] Steve Lohr. Potboiler springs from computer's loins. *New York Times*, page A1, Jul 1993.

[700] Birte Lönneker. Narratological knowledge for natural language generation. In *Proc. European Workshop on Natural Language Generation*, 2005.

[701] Sandy Louchart and Ruth Aylett. Emergent narrative, requirements and high-level architecture. In *Proc. Hellenic Conference on Artificial Intelligence*, pages 298–308, 2004.

[702] Sandy Louchart and Ruth Aylett. The emergent narrative theoretical investigation. In *Proc. Narrative and Interactive Learning Environments*, pages 21–28, 2004.

[703] Sandy Louchart and Ruth Aylett. Narrative theory and emergent interactive narrative. *International Journal of Continuing Engineering Education and Life Long Learning*, 14(6):506–518, 2004.

[704] Sandy Louchart, Ruth Aylett, Michael Kriegel, João Dias, Rui Figueiredo, and Ana Paiva. Authoring emergent narrative-based games. *Journal of Game Development*, 3(1):19–37, 2008.

[705] Sandy Louchart, Ivo Swartjes, Michael Kriegel, and Ruth Aylett. Purposeful authoring for emergent narrative. In *Proc. Joint International Conference on Interactive Digital Storytelling*, pages 273–284. Springer, 2008.

[706] Henry Lowood. Storyline, dance/music, or PvP? Game movies and community players in World of Warcraft. *Games and Culture*, 1(4):362–382, 2006.

[707] A Bryan Loyall and Joseph Bates. Personality-rich believable agents that use language. In *Proc. Autonomous Agents*, 1997.

[708] A Bryan Loyall, W Reilly, Joseph Bates, and Peter Weyhrauch. System for authoring highly interactive, personality-rich interactive characters. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 59–68. Eurographics Association, 2004.

[709] Bryan Loyall. Response to Janet Murray's 'from game-story to cyberdrama'. *First person: New media as story, performance, and game*, 1:2–9, 2004.

[710] Bryan Loyall, A Bryan, and Loyall Joseph Bates. Hap a reactive, adaptive architecture for agents. Technical Report CMU-CS-91-147, Carnegie Mellon University, 1991.

[711] David P Luebke. *Level of detail for 3D graphics*. Morgan Kaufmann, 2003.

[712] Stephanie Lukin, Lena Reed, and Marilyn Walker. Generating sentence planning variations for story telling. In *Proc. Special Interest Group on Discourse and Dialogue*, pages 188–197, 2015.

[713] Stephanie M Lukin. *Generating variations in a virtual storyteller*. PhD thesis, University of California, Santa Cruz, 2017.

[714] Stephanie M Lukin, Kevin Bowden, Casey Barackman, and Marilyn A Walker. Personabank: A corpus of personal narratives and their story intention graphs. In *Proc. Language Resources*, 2017.

[715] Stephanie M Lukin, James O Ryan, and Marilyn A Walker. Automating direct speech variations in stories and games. In *Proc. Games and Natural Language Processing*, 2014.

[716] Stephanie M Lukin and Marilyn A Walker. Narrative variations in a virtual storyteller. In *Proc. International Conference on Intelligent Virtual Agents*, pages 320–331. Springer, 2015.

[717] Shanhong Luo and Guangjian Zhang. What leads to romantic attraction: similarity, reciprocity, security, or beauty? evidence from a speed-dating study. *Journal of Personality*, 77(4):933–964, 2009.

[718] Deborah Lupton. *The quantified self*. John Wiley & Sons, 2016.

[719] Theo Lutz. Stochastische texte. *Augenblick*, 4:3–9, 1959.

[720] John M MacGregor. *The discovery of the art of the insane*. Princeton University Press, 1989.

[721] Josephine Machon. *Immersive theatres: Intimacy and immediacy in contemporary performance*. Palgrave Macmillan, 2013.

[722] David Maclagan. *Outsider art: From the margins to the marketplace*. Reaktion Books, 2010.

[723] Pattie Maes. How to do the right thing. *Connection Science*, 1(3):291–323, 1989.

[724] Brian Magerko. *Player modeling in the interactive drama architecture*. PhD thesis, University of Michigan, 2006.

[725] Brian Magerko, Christopher DeLeon, and Peter Dohogne. Digital improvisational theatre: Party Quirks. In *Proc. International Workshop on Intelligent Virtual Agents*, pages 42–47. Springer, 2011.

[726] Brian Magerko, Peter Dohogne, and Christopher DeLeon. Employing fuzzy concept for digital improvisational theatre. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2011.

[727] François Mairesse. *Learning to adapt in dialogue systems: data-driven models for personality recognition and generation*. PhD thesis, University of Sheffield, 2008.

[728] François Mairesse and Marilyn Walker. PERSONAGE: Personality generation for dialogue. In *Proc. Association of Computational Linguistics*, pages 496–503, 2007.

[729] François Mairesse and Marilyn A Walker. A personality-based framework for utterance generation in dialogue applications. In *Proc. AAAI Spring Symposium on Emotion, Personality, and Social Behavior*, pages 80–87, 2008.

[730] François Mairesse and Marilyn A Walker. Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*, 37(3):455–488, 2011.

[731] Robert C Malenka and Roger A Nicoll. Long-term potentiation—a decade of progress? *Science*, 285(5435):1870–1874, 1999.

[732] Jeff Mallett, Matthias Wüllenweber, and Frans Morsch. The 2010 Herschberg best-annotation award. *ICGA Journal*, page 56, 2010.

[733] George Mandler. The structure of value: Accounting for taste. *Center for Human Information Processing Report*, 1982.

[734] William Mann. Text generation. *Computational Linguistics*, 8(2):62–69, 1982.

[735] Zohar Manna and Richard Waldinger. A deductive approach to program synthesis. In *Readings in Artificial Intelligence and Software Engineering*, pages 3–34. Elsevier, 1986.

[736] Zohar Manna and Richard J Waldinger. Toward automatic program synthesis. *Communications of the ACM*, 14(3):151–165, 1971.

[737] Jess Marcotte. GGJ 2017: Some thoughts on rustle your leaves to me softly. `http://tag.hexagram.ca/jekagames/ggj-2017-some-thoughts-on-rustle-your-leaves-to-me-softly/`, 2017.

[738] Jess Marcotte. The Truly Terrific Traveling Troubleshooter at QGCon 2017. `http://tag.hexagram.ca/jekagames/the-truly-terrific-traveling-troubleshooter-at-qgcon-2017/`, 2017.

[739] Piotr Marecki and Nick Montfort. Renderings: Translating literary works in the digital age. *Digital Scholarship in the Humanities*, 32:i84–i91, 2017.

[740] Mark Marino and Rob Wittig. Netprov: Elements of an emerging form. *Dichtung Digital*, 42, 2012.

[741] Dave Mark. AIIDE 2009—AI challenges in Sims 3—Richard Evans. `http://intrinsicalgorithm.com/IAonAI/2009/10/aiide-2009-ai-challenges-in-sims-3-richard-evans/`, Oct 2009. Last accessed: December 13, 2018.

[742] Dave Mark. *Behavioral Mathematics for Game AI.* Cengage Learning, 2009.

[743] Keith D Markman, William MP Klein, and Julie A Suhr. *Handbook of imagination and mental simulation.* Psychology Press, 2012.

[744] George Marsaglia. Random number generation. In Anthony Ralston, Edwin D. Reilly, and David Hemmendinger, editors, *Encyclopedia of Computer Science*. John Wiley and Sons, Ltd., 2003.

[745] Stacy C Marsella and Jonathan Gratch. Ema: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70–90, 2009.

[746] Stacy C Marsella, David V Pynadath, and Stephen J Read. Psychsim: Agent-based modeling of social interactions and influence. In *Proc. International Conference on Cognitive Modeling*, pages 243–248, 2004.

[747] TA Marsland and Y Björnsson. Fritz 5.32 wins the 1998 Herschberg best-annotation award. *ICGA Journal*, 22(2):116–118, 1999.

[748] Chris Martens. Towards computational support for experimental theater. http://www.cs.cmu.edu/~cmartens/theater-tools-abstract.pdf, 2016.

[749] Chris Martens and Rogelio E Cardona-Rivera. Discourse-driven comic generation. In *Proc. International Conference on Computational Creativity*, 2016.

[750] Chris Martens and Rogelio E Cardona-Rivera. Generating abstract comics. In *Proc. International Conference on Interactive Digital Storytelling*, pages 168–175. Springer, 2016.

[751] Paul Martin. Landscape and gamescape in Dwarf Fortress. In *Proc. Philosophy of Computer Games*, 2012.

[752] Abraham H Maslow. A theory of human motivation. *Psychological Review*, 50(4):370, 1943.

[753] Stacey Mason. On games and links: Extending the vocabulary of agency and immersion in interactive narratives. In *Proc. International Conference on Interactive Digital Storytelling*, pages 25–34. Springer, 2013.

[754] Zackery Mason. Historic computer simulation. Technical Report IQP MBJ 1604, Worcester Polytechnic Institute, 2016. Last accessed: May 17, 2018.

[755] Margaret Masterman. Semantic message detection for machine translation, using an interlingua. In *Proc. International Conference on Machine Translation*, pages 438–475, 1961.

[756] Gus Mastrapa. Kiwi comic tells tale of Dwarf Fortress failure. *Wired*, Sep 2010.

[757] Michael Mateas. Computational subjectivity in virtual world avatars. In *Proc. AAAI Fall Symposium on Socially Intelligent Agents Symposium*, 1997.

[758] Michael Mateas. An Oz-centric review of interactive drama and believable agents. Technical Report CMUâĹŠCSâĹŠ97âĹŠ156, Carnegie Mellon University, 1997.

[759] Michael Mateas. Subjective avatars. In *Proc. International Conference on Autonomous Agents*, 1998.

[760] Michael Mateas. Not your grandmother's game: AI-based art and entertainment. In *Proc. AAAI Spring Symposium on Artificial Intelligence and Computer Games*, 1999.

[761] Michael Mateas. A neo-Aristotelian theory of interactive drama. In *Proc. AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, 2000.

[762] Michael Mateas. Expressive ai: A hybrid art and science practice. *Leonardo*, 34(2):147–153, 2001.

[763] Michael Mateas. A preliminary poetics for interactive drama and games. *Digital Creativity*, 12(3):140–152, 2001.

[764] Michael Mateas. *Interactive drama, art and artificial intelligence*. PhD thesis, Carnegie Mellon University, 2002.

[765] Michael Mateas. Expressive AI: A semiotic analysis of machinic affordances. In *Proc. Computational Semiotics for Games and New Media*, pages 58–67, 2003.

[766] Michael Mateas. Expressive ai: A semiotic analysis of machinic affordances. In *Proc. Computational Semiotics for Games and New Media*, pages 58–67, 2003.

[767] Michael Mateas. Build it to understand it: Ludology meets narratology in game design space. 2005.

[768] Michael Mateas, Steffi Domike, and Paul Vanouse. Terminal time: An ideologically-biased history machine. *AISB Quarterly*, 102:36–43, 1999.

[769] Michael Mateas, Peter A Mawhorter, and Noah Wardrip-Fruin. Intentionally generating choices in interactive narratives. In *Proc. International Conference on Computational Creativity*, pages 292–299, 2015.

[770] Michael Mateas and Josh McCoy. An architecture for character-rich social simulation. *Game AI Pro: Collected Wisdom of Game AI Professionals*, pages 515–530, 2013.

[771] Michael Mateas and Nick Montfort. A box, darkly: Obfuscation, weird languages, and code aesthetics. In *Proc. Digital Arts and Culture Conference*, pages 144–153, 2005.

[772] Michael Mateas and Phoebe Sengers. *Narrative intelligence*. J. Benjamins Pub., 2003.

[773] Michael Mateas and Andrew Stern. A behavior language for story-based believable agents. *IEEE Intelligent Systems*, 17(4):39–47, 2002.

[774] Michael Mateas and Andrew Stern. Towards integrating plot and character for interactive drama. In *Socially Intelligent Agents*, pages 221–228. Springer, 2002.

[775] Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference*, 2003.

[776] Michael Mateas and Andrew Stern. Integrating plot, character and natural language processing in the interactive drama façade. In *Proc. International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2003.

[777] Michael Mateas and Andrew Stern. A behavior language: Joint action and behavioral idioms. In Helmut Prendinger and Mitsuru Ishizuka, editors, *Life-Like Characters*, pages 135–161. Springer, 2004.

[778] Michael Mateas and Andrew Stern. Structuring content in the Façade interactive drama architecture. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, pages 93–98, 2005.

[779] Michael Mateas and Andrew Stern. Writing façade: A case study in procedural authorship. In Pat Harrigan and Noah Wardrip-Fruin, editors, *Second person: Role-playing and story in games and playable media*, pages 183–208. 2007.

[780] Michael Mateas, Paul Vanouse, and Steffi Domike. Generation of ideologically-biased historical documentaries. In *Proc. National Conference on Artificial Intelligence*, pages 236–242, 2000.

[781] Michael Mateas and Noah Wardrip-Fruin. Defining operational logics. 2009.

[782] Michael Mateas and Noah Wardrip-Fruin. Personalized and interactive literature. In William S Bainbridge and Mihail C Roco, editors, *Handbook of Science and Technology Convergence*. Springer, 2016.

[783] Peter Mawhorter. Reader-model-based story generation. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, 2013.

[784] Peter Mawhorter. Efficiency, realism, and representation in generated content: a case study using family tree generation. In *Proc. Procedural Content Generation*. ACM, 2017.

[785] Peter Mawhorter, Michael Mateas, and Noah Wardrip-Fruin. Generating relaxed, obvious, and dilemma choices with dunyazad. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, pages 58–64, 2015.

[786] Peter Mawhorter, Michael Mateas, Noah Wardrip-Fruin, and Arnav Jhala. Towards a theory of choice poetics. In *Proc. International Conference on the Foundations of Digital Games*. 2014.

[787] Peter Mawhorter, Carmen Zegura, Alex Gray, Arnav Jhala, Michael Mateas, and Noah Wardrip-Fruin. Choice poetics by example. In *Arts*, volume 7. Multidisciplinary Digital Publishing Institute, 2018.

[788] Peter A Mawhorter. *Artificial intelligence as a tool for understanding narrative choices*. PhD thesis, University of California, Santa Cruz, 2016.

[789] Peter A Mawhorter. Dunyazad Git repository. `https://github.com/solsword/dunyazad`, 2016.

[790] Manfred Max-Neef. Development and human needs. In Paul Ekins and Manfred A Max-Neef, editors, *Real-life economics: Understanding wealth creation*, pages 197–214. Routledge, 1992.

[791] Maxis. *SimCity*. Maxis, 1989.

[792] Maxis. *The Sims*. Electronic Arts, 2000.

[793] Maxis. *The Sims 2*. Electronic Arts, 2004.

[794] Dan P McAdams. Personal narratives and the life story. *Handbook of Personality: Theory and Research*, pages 242–262, 2008.

[795] Jim McClellan. Hooray for Silliwood: Jim McClellan looks at the growing video game movie industry. *The Observer*, pages 84–85, Nov 1994.

[796] Scott McCloud. *Understanding Comics: The Invisible Art*. Tundra Publishing, 1993.

[797] Josh McCoy. Simulations of interactions through a game model. Bachelor's thesis, Earlham College, Apr 2004.

[798] Josh McCoy, Michael Mateas, Derek H Hart, Jonathan Whetzel, Justin Derrick Basilico, Matthew R Glickman, and Robert G Abbott. Enabling immersive simulation. Technical Report SAND2009-0840, Sandia National Laboratories, 2009.

[799] Josh McCoy, Mike Treanor, Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. Prom Week: Social physics as gameplay. In *Proc. International Conference on the Foundations of Digital Games*, 2011.

[800] Josh McCoy, Mike Treanor, Ben Samuel, Brandon Tearse, Michael Mateas, and Noah Wardrip-Fruin. Authoring game-based interactive narrative using social games and Comme il Faut. In *Proc. International Conference & Festival of the Electronic Literature Organization*, 2010.

[801] Josh McCoy, Mike Treanor, Ben Samuel, Brandon Tearse, Michael Mateas, and Noah Wardrip-Fruin. Comme il faut 2: A fully realized model for socially-oriented gameplay. In *Proc. Intelligent Narrative Technologies III*. ACM, 2010.

[802] Joshua McCoy and Michael Mateas. The computation of self in everyday life: A dramaturgical approach for socially competent agents. In *Proc. AAAI Spring Symposium: Intelligent Narrative Technologies II*, pages 75–82, 2009.

[803] Joshua McCoy, Mike Treanor, Ben Samuel, Aaron A Reed, Michael Mateas, and Noah Wardrip-Fruin. Prom Week: Designing past the game/story dilemma. In *Proc. International Conference on the Foundations of Digital Games*, pages 94–101, 2013.

[804] Joshua McCoy, Mike Treanor, Ben Samuel, Aaron A Reed, Michael Mateas, and Noah Wardrip-Fruin. Social story worlds with Comme il Faut. *Transactions on Computational Intelligence and AI in Games*, 6(2):97–112, 2014.

[805] Joshua McCoy, Mike Treanor, Ben Samuel, Brandon Robert Tearse, Michael Mateas, and Noah Wardrip-Fruin. The prom: An example of socially-oriented gameplay. In *AIIDE*, 2010.

[806] Joshua McCoy, Mike Treanor, Ben Samuel, Noah Wardrip-Fruin, and Michael Mateas. Comme il Faut: A system for authoring playable social models. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, 2011.

[807] Joshua Allen McCoy. *All the world's a stage: A playable model of social interaction inspired by dramaturgical analysis*. PhD thesis, University of California, Santa Cruz, 2012.

[808] Daniel D McCracken and Edwin D Reilly. Backus-Naur form (bnf). 2003.

[809] Robert R McCrae and Paul T Costa. The structure of interpersonal traits: Wiggins's circumplex and the five-factor model. *Journal of Personality and Social Psychology*, 56(4):586, 1989.

[810] Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of Personality*, 60(2):175–215, 1992.

[811] David D McDonald. Mumble: A flexible system for language production. In *Proc. International Joint Conference on Artificial Intelligence*, pages 1062–1062. Morgan Kaufmann Publishers, Inc., 1981.

[812] David D McDonald. Description directed control: Its implications for natural language generation. *Computational Linguistics*, 5:111–129, 1983.

[813] David D McDonald. `https://github.com/ddmcdonald/sparser/tree/master/Mumble`, 2018. Last accessed: November 17, 2018.

[814] David D McDonald and James Pustejovsky. Description-directed natural language generation. In *Proc. International Joint Conference on Artificial Intelligence*, pages 799–805, 1985.

[815] David Daniel McDonald. *Natural language production as a process of decision-making under constraints*. PhD thesis, Massachusetts Institute of Technology, 1980.

[816] Neil McIntyre and Mirella Lapata. Learning to tell tales: A data-driven approach to story generation. In *Proc. Joint Conference of Association for Computational Linguistics and the International Conference on Natural Language Processing*, pages 217–225. Association for Computational Linguistics, 2009.

[817] Kevin McKean. Computers, fiction, and poetry. *Byte*, 7(7), Jul 1982.

[818] Robert McKee. *Story: Substance, structure, style, and the principles of screenwriting*. Harper Collins, 1997.

[819] Kathleen McKeown. *Text generation*. Cambridge University Press, 1992.

[820] Gordon MeCalla and Peter F Schneider. The execution of plans in an independent dynamic microworld. In *Proc. International Joint Conference on Artificial Intelligence*, pages 553–555. Morgan Kaufmann Publishers Inc., 1979.

[821] James R Meehan. Using planning structures to generate stories. *American Journal of Computational Linguistics*, 33:78–94, 1975.

[822] James R Meehan. *The Metanovel: writing stories by computer*. PhD thesis, Yale University, 1976.

[823] James R Meehan. Tale-Spin, an interactive program that writes stories. In *Proc. International Joint Conference on Artificial Intelligence*, pages 91–98, 1977.

[824] James R Meehan. Stories and cognition: Comments on Robert de Beaugrande's "the story of grammars and the grammar of stories". *Journal of Pragmatics*, 6(5-6):455–462, 1982.

[825] James Richard Meehan. Boy meets goal, boy loses goal, boy gets goal: The nature of feedback between goal-based simulation and understanding systems. Technical Report Technical Report 170, University of California, Irvine, 1981.

[826] Elijah Meeks. An interview with Tarn Adams. `https://dhs.stanford.edu/algorithmic-literacy/an-interview-with-tarn-adams`, Dec 2010.

[827] Eleazar Meletinsky, Sergej Nekludov, Elena Novik, and Dimitri Segal. Problems of the structural analysis of fairytales. *Soviet Structural Folkloristics*, 1:73–139, 1974.

[828] Melpomene. *Bagabone, Hem'I Die Now*. Vantage Press, 1980.

[829] Herman Melville. *Moby-Dick; or, The Whale*. Project Gutenberg, 2017.

[830] Gonzalo Méndez, Pablo Gervás, and Carlos León. A model of character affinity for agent-based story generation. In *Proc. Knowledge, Information and Creativity Support Systems*, 2014.

[831] Gonzalo Méndez, Pablo Gervás, and Carlos León. On the use of character affinities for story plot generation. In *Proc. Knowledge, Information and Creativity Support Systems*, pages 211–225. Springer, 2016.

[832] Rosa Menkman. Glitch studies manifesto. In Geert Lovink and Rachel Somers Miles, editors, *Video Vortex Reader II: Moving Images Beyond YouTube*, pages 336–347. 2011.

[833] Roger Midmorn. An interpretation of Sheldon Klein's four valued analogical transformational operator. Technical Report TR1801, University of Wisconsin, 2014.

[834] China Miéville. China miéville on crime novels. *Whatever*, May 2009.

[835] China Mieville. There and back again: Five reasons Tolkien rocks. *Omnivoracious*, June 2009.

[836] Louis T Milic. The possible usefulness of poetry generation. 1970.

[837] Robert McColl Millar and Larry Trask. *Trask's historical linguistics*. Routledge, 2015.

[838] James Milroy and Lesley Milroy. Linguistic change, social network and speaker innovation. *Journal of Linguistics*, 1985.

[839] Kent Minturn. Dubuffet, lévi-strauss, and the idea of art brut. *RES: Anthropology and Aesthetics*, 46(1):247–258, 2004.

[840] Cheryl Misak. C. S. Peirce on vital matters. In Cheryl Misak, editor, *The Cambridge Companion to Peirce*. Cambridge University Press, 2004.

[841] Alex Mitchell. Reflective rereading and the simcity effect in interactive stories. In *Proc. International Conference on Interactive Digital Storytelling*, pages 27–39. Springer, 2015.

[842] Satya P Mohanty and Satya Mohanty. *Literary theory and the claims of history: Postmodernism, objectivity, multicultural politics*. Cornell University Press, 1997.

[843] Henry Mohr, Markus Eger, and Chris Martens. Eliminating the impossible: A procedurally generated murder mystery. In *Proc. Experimental AI in Games*, 2018.

[844] Herbert Molderings. *Duchamp and the Aesthetics of Chance: Art as Experiment*. Columbia University Press, 2010.

[845] Monolith Productions. *Middle-earth: Shadow of Mordor*. Warner Bros. Interactive Entertainment, 2014.

[846] Nick Montfort. *Twisty Little Passages: an approach to interactive fiction*. MIT Press, 2005.

[847] Nick Montfort. *Generating narrative variation in interactive fiction*. PhD thesis, University of Pennsylvania, 2007.

[848] Nick Montfort. The ppg256 series of minimal poetry generators. In *Proc. Digital Arts and Culture*, 2009.

[849] Nick Montfort. Curveship's automatic narrative style. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 211–218. ACM, 2011.

[850] Nick Montfort and Natalia Fedorova. Small-scale systems and computational creativity. In *Proc. International Conference on Computational Creativity*, page 82, 2012.

[851] Nick Montfort, Rafael Pérez y Pérez, D Fox Harrell, and Andrew Campana. Slant: A blackboard system to generate plot, figuration, and narrative discourse aspects of stories. In *Proc. International Conference on Computational Creativity*, pages 168–175, 2013.

[852] Nick Montfort and Ariane Savoie. 'Exquisite Corpses' unearthed from French newspaper. In *Proc. History of Expressive Systems*. In press.

[853] Nick Montfort, Erik Stayton, and Andrew Campana. Expressing the narrator's expectations. In *Proc. Intelligent Narrative Technologies*, 2014.

[854] SungRim Moon, Serguei Pakhomov, James Ryan, and Genevieve B Melton. Automated non-alphanumeric symbol resolution in clinical texts. In *Proc. American Medical Informatics Association Annual Symposium*, volume 2011, page 979. American Medical Informatics Association, 2011.

[855] Masahiro Mori. The uncanny valley. *Energy*, 7(4):33–35, 1970.

[856] Brian Moriarty. I sing the story electric. `http://ludix.com/moriarty/electric.html`, 2015. Last accessed: May 17, 2018.

[857] Jason Morningstar. *Fiasco*. Bully Pulpit Games, 2009.

[858] Billy Morrissette. *Scotland, PA*. Lot 47 Films, 2001.

[859] Harrison R Morse. Preliminary operating notes for SAGA II. Technical Report 8436-M-29 (Memorandum), Massachusetts Institute of Technology, 1960.

[860] Mountain Morning Show. Bad News. `https://www.youtube.com/watch?v=gaSTV0gJv14`, Jan 2017.

[861] Erik Mueller. An update on DAYDREAMER, a computer model of human daydreaming. In *Proc. History of Expressive Systems*. In press.

[862] Erik T Mueller. *Daydreaming in humans and machines: a computer model of the stream of thought*. Intellect Books, 1990.

[863] Erik T Mueller, Michael G Dyer, et al. Daydreaming in humans and computers. In *Proc. International Joint Conference on Artificial Intelligence*, pages 278–280, 1985.

[864] Samuel Munilla and R Michael Young. Zuzen, a cloud-based framework for automated machinima generation. In *Proc. International Conference on Advances in Computer Enterntainment Technology*, pages 199–206. ACM, 2009.

[865] Dane Munro. Memento mori and vanitas elements in the funerary art at St John's co-cathedral, valletta, malta. *Treasures of Malta*, 11, 2004.

[866] Eric Murnane. Three Hundred Days in the North. `https://tandtprojects.cah.ucf.edu/~er397450/300days.html`, 2017. Last accessed: June 14, 2018.

[867] Eric Murnane. *Emergent Narrative: Stories of Play, Playing with Stories.* PhD thesis, University of Central Florida, 2018.

[868] Janet Murray. Dr. Quinn on the Holodeck, or blueprint for an electronic storyland. In *Working Notes of the AAAI Spring Symposium on Interactive Story Systems*, 1995.

[869] Janet H Murray. The last word on ludology v narratology in game studies. In *Proc. Digital Games Research Association*, 2005.

[870] Janet Horowitz Murray and Janet H Murray. *Hamlet on the Holodeck: The future of narrative in cyberspace.* MIT Press, 1997.

[871] Michael Murray. Narrative psychology. In Jonathan A Smith, editor, *Qualitative Psychology: A Practical Guide to Research Methods*, pages 111–131. 2003.

[872] Michael Neff, Yingying Wang, Rob Abbott, and Marilyn Walker. Evaluating the effect of gesture and language on personality perception in conversational agents. In *International Conference on Intelligent Virtual Agents*, pages 222–235. Springer, 2010.

[873] Chrystopher Nehaniv and Kerstin Dautenhahn. Embodiment and memories—algebras of time and history for autobiographic agents. *Cybernetics and Systems*, 98:651–656, 1998.

[874] Chrystopher L Nehaniv. What's your story? irreversibility, algebra, autobiographic agents. In *Proc. Socially Intelligent Agents)*, 1997.

[875] Chrystopher L Nehaniv and Kerstin Dautenhahn. Semigroup expansions for autobiographic agents. In *Proc. Symposium on Algebra, Languages and Computation*, pages 77–84, 1998.

[876] Theodore H Nelson. Computer lib/dream machines. 1987.

[877] Kenneth B Newell. Pattern, concrete, and computer poetry: The poem as object in itself. *The Bucknell Review*, 27(2):159, 1983.

[878] Henrik Skov Nielsen, James Phelan, and Richard Walsh. Ten theses about fictionality. *Narrative*, 23(1):61–73, 2015.

[879] Nintendo Creative Department. *Super Mario Bros.* Nintendo, 1985.

[880] Nintendo EAD. *The Legend of Zelda: Majora's Mask*. Nintendo, 2000.

[881] Nintendo EAD. *The Legend of Zelda: The Wind Waker*. Nintendo, 2003.

[882] Neal R Norrick. The dark side of tellability. *Narrative Inquiry*, 15(2):323–343, 2005.

[883] Rockstar North. *Grand Theft Auto: Vice City*. Rockstar Games, 2002.

[884] Nuno Nunes, Ian Oakley, and Valentina Nisi. *Interactive Storytelling: Proceedings of the 10th International Conference on Interactive Digital Storytelling, ICIDS 2017 Funchal, Madeira, Portugal, November 14–17, 2017*. Springer, 2017.

[885] David E Nye. *American technological sublime*. MIT Press, 1996.

[886] Thomas Patrick Oates. New media and the repackaging of NFL fandom. *Sociology of Sport Journal*, 26(1):31–49, 2009.

[887] Keith Oatley. *Such stuff as dreams: The psychology of fiction*. John Wiley & Sons, 2011.

[888] Sensible Object. *When in Rome*. Sensible Object, 2018.

[889] Sensible Object. *Word Mine*. Sensible Object, 2018.

[890] Elinor Ochs and Lisa Capps. Narrating the self. *Annual Review of Anthropology*, 25(1):19–43, 1996.

[891] Elinor Ochs and Lisa Capps. *Living narrative: Creating lives in everyday storytelling*. Harvard University Press, 2001.

[892] Patrick O'Halloran. A multi-tiered level of detail system for game AI. Master's thesis, University of Dublin, Trinity College, 2016.

[893] Richard Ohmann. Speech acts and the definition of literature. *Philosophy & Rhetoric*, pages 1–19, 1971.

[894] Yuichi Ohta and Hideyuki Tamura. *Mixed reality: merging real and virtual worlds*. Springer, 2014.

[895] Arika Okrent. *In the land of invented languages: Esperanto rock stars, Klingon poets, Loglan lovers, and the mad dreamers who tried to build a perfect language*. Spiegel & Grau, 2009.

[896] Ramon Oldenburg and Dennis Brissett. The third place. *Qualitative Sociology*, 5(4):265–284, 1982.

[897] Ray Oldenburg. *Celebrating the third place: Inspiring stories about the great good places at the heart of our communities*. Da Capo Press, 2001.

[898] OldThinkerTube. Façade—how to run away from Trip 2017 update. https://www.youtube.com/watch?v=4g14ceDolPQ, May 2017. Last accessed: May 15, 2018.

[899] Martin O'Leary. Generating naming languages. https://mewo2.com/notes/naming-language, 2015. Last accessed: December 14, 2018.

[900] Martin O'Leary. *Uncharted Atlas*. 2015.

[901] Melvin Oliver and Thomas Shapiro. *Black wealth/white wealth: A new perspective on racial inequality*. Routledge, 2013.

[902] David Olsen and Mark J Nelson. The narrative logic of rube goldberg machines. In *Proc. International Conference on Interactive Digital Storytelling*, pages 104–116. Springer, 2017.

[903] Brian O'Neill, Andreya Piplica, Daniel Fuller, and Brian Magerko. A knowledge-based framework for the collaborative improvisation of scene introductions. In *Proc. International Conference on Interactive Digital Storytelling*, pages 85–96. Springer, 2011.

[904] Jeff Orkin. Three states and a plan: the AI of FEAR. In *Game Developers Conference*, 2006.

[905] Charles L Ortiz Jr. Explanatory update theory: Applications of counterfactual reasoning to causation. *Artificial Intelligence*, 108(1-2):125–178, 1999.

[906] Joseph Osborn, Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. Playspecs: Regular expressions for game play traces. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, pages 170–176, 2015.

[907] Joseph C. Osborn, James Ryan, and Michael Mateas. Analyzing expressionist grammars by reduction to symbolic visibly pushdown automata. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2017.

[908] Joseph C Osborn, Ben Samuel, Adam Summerville, and Michael Mateas. Towards general RPG playing. In *Proc. Experimental AI in Games*, pages 92–98, 2017.

[909] Joseph Carter Osborn. *Operationalizing Operational Logics.* PhD thesis, University of California, Santa Cruz, 2018.

[910] Joseph Carter Osborn, Ben Samuel, Joshua Allen McCoy, and Michael Mateas. Evaluating play trace (dis)similarity metrics. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2014.

[911] CS Pacat. Fantasy worldbuilding: The power of detail. *Writing Queensland*, (258):14, 2017.

[912] Rafael R Padovani, Lucas N Ferreira, and Levi HS Lelis. Bardo: Emotion-based music recommendation for tabletop role-playing games. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2017.

[913] Alan Palmer. *Fictional minds.* University of Nebraska Press, 2004.

[914] Frederick Palmer. Photoplay plot encyclopedia, 1922.

[915] Jo Alyson Parker. *Narrative form and chaos theory in Sterne, Proust, Woolf, and Faulkner.* Springer, 2007.

[916] JR Parker and Katrin Becker. The simulation-game controversy: What is a ludic simulation? *International Journal of Gaming and Computer-Mediated Simulations*, 5(1):1–12, 2013.

[917] Rob Parker. The culture of permadeath: Roguelikes and terror management theory. *Journal of Gaming & Virtual Worlds*, 9(2):123–141, 2017.

[918] Terence Parsons. A meinongian analysis of fictional objects. *Grazer Philosophische Studien*, 1:73–86, 1975.

[919] Richard Paul, Darryl Charles, Michael McNeill, and David McSherry. Adaptive storytelling and story repair in a dynamic environment. In *Proc. International Conference on Interactive Digital Storytelling*, pages 128–139. Springer, 2011.

[920] James Paul Gee. The narrativization of experience in the oral style. *Journal of Education*, 167(1):9–35, 1985.

[921] Thomas G Pavel. *Fictional worlds.* Harvard University Press, 1986.

[922] Ade Peace. The city and the circus: engagement, symbol and drama in the Adelaide Grand Prix. pages 95–115, 1991.

[923] Celia Pearce. Towards a game theory of game. *First person: New media as story, performance, and game*, pages 143–153, 2004.

[924] Celia Pearce. Narrative environments. *Space Time Play*, pages 200–205, 2007.

[925] Federico Peinado. Interactive digital storytelling: Automatic direction of virtual environments. *Upgrade. Monograph: Virtual Environments*, 7(2):42–46, 2006.

[926] Federico Peinado and Pablo Gervás. Transferring game mastering laws to interactive digital storytelling. In *Proc. International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pages 48–54. Springer, 2004.

[927] Lucienne Peiry. *Art brut: The origins of outsider art*. Flammarion-Pere Castor, 2001.

[928] Lyn Pemberton. A modular approach to story generation. In *Proc. Conference of the European Chapter of the Association for Computational Linguistics*, pages 217–224. Association for Computational Linguistics, 1989.

[929] Stephen C Pepper. *Concept and quality: A world hypothesis*. Open Court, 1966.

[930] Stephen Coburn Pepper. *World hypotheses: A study in evidence*. University of California Press, 1942.

[931] Rafael Pérez y Pérez. *MEXICA: A Computer Model of Creativity in Writing*. PhD thesis, University of Sussex, 1999.

[932] Rafael Pérez y Pérez. Employing emotions to drive plot generation in a computer-based storyteller. *Cognitive Systems Research*, 8(2):89–109, 2007.

[933] Rafael Pérez y Pérez. From MEXICA to MEXICA-Impro: The evolution of a computer model for plot generation. In *Computational Creativity Research: Towards Creative Machines*, pages 267–284. Springer, 2015.

[934] Rafael Pérez y Pérez. *Mexica: 20 Years—20 Stories [20 años—20 historias]*. Counterpath Press, 2017.

[935] Rafael Pérez y Pérez, Nora Morales, and Luis Rodríguez. Illustrating a computer generated narrative. In *ICCC*, pages 103–110, 2012.

[936] Rafael Pérez y Pérez and Mike Sharples. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13(2):119–139, 2001.

[937] Rafael Pérez y Pérez and Mike Sharples. Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-Based Systems*, 17(1):15–29, 2004.

[938] Rafael Pérez y Pérez, Ricardo Sosa, and Christian Lemaitre. A computer model for visual-daydreaming. In *Proc. Intelligent Narrative Technologies*, 2007.

[939] Ken Perlin and Athomas Goldberg. Improv: A system for scripting interactive actors in virtual worlds. In *Proc. Computer graphics and interactive techniques*, pages 205–216. ACM, 1996.

[940] Alan Peshkin. In search of subjectivity—one's own. *Educational Researcher*, 17(7):17–21, 1988.

[941] Amanda Phillips, Gillian Smith, Michael Cook, and Tanya Short. Feminism and procedural content generation: toward a collaborative politics of computational creativity. *Digital Creativity*, 27(1):82–97, 2016.

[942] Matthew Pike, Richard Ramchurn, Steve Benford, and Max L Wilson. #scanners: Exploring the control of adaptive films using brain-computer interaction. In *Proc. ACM CHI Conference on Human Factors in Computing Systems*, pages 5385–5396. ACM, 2016.

[943] Matthew Pike, Richard Ramchurn, and Max L Wilson. #scanners: integrating physiology into cinematic experiences. In *Proc. ACM SIGCHI Conference on Creativity and Cognition*, pages 151–152. ACM, 2015.

[944] Matthew Pike, Richard Ramchurn, and Max L Wilson. Two-way affect loops in multimedia experiences. In *Proc. British HCI Conference*, pages 117–118. ACM, 2015.

[945] Matthew Pike, Max L Wilson, Steve Benford, and Richard Ramchurn. #scanners: A bci enhanced cinematic experience. In *Proc. ACM CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 293–296. ACM, 2016.

[946] Tomáš Plch. *Believable decision making in large scale open world games for ambient characters*. PhD thesis, Charles University in Prague, 2018.

[947] Robert Plomin. *Genetics and experience: The interplay between nature and nurture*. Sage, 1994.

[948] Robert K Plumb. Scholars praise computers' uses. *New York Times*, Sep 1964.

[949] Livia Polanyi. So what's the point? *Semiotica*, 25(3-4):207–242, 1979.

[950] Georges Polti. *The Thirty-six Dramatic Situations*. Editor Company, 1916.

[951] Jason Porterfield. *The Homestead Act of 1862: A Primary Source History of the Settlement of the American Heartland in the Late 19th Century*. The Rosen Publishing Group, 2004.

[952] James Portnow. Gaming languages and language games. In Michael Adams, editor, *From Elvish to Klingon: Exploring Invented Languages*, pages 135–159. Oxford University Press, 2011.

[953] Emil L Post. Finite combinatory processes—formulation 1. *The Journal of Symbolic Logic*, 1(3):103–105, 1936.

[954] Emil L Post. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65(2):197–215, 1943.

[955] Mark Allan Powell. *What is narrative criticism?* Fortress Press, 1990.

[956] Mary Louise Pratt. Toward a speech act theory of literary discourse. 1977.

[957] Alex Preminger, OB Hardison Jr, and Frank J Warnke. *The Princeton handbook of poetic terms*. Princeton University Press, 2014.

[958] Alan Prince and Paul Smolensky. Optimality theory: Constraint interaction in generative grammar. Technical Report Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado Boulder, 1993.

[959] Gerald Prince. The disnarrated. *Style*, pages 1–8, 1988.

[960] Vladimir Propp. *Morphology of the Folktale*. University of Texas Press, 2010 [1928].

[961] SF Luthfie Arguby Purnomo, Mangatur Nababan, Riyadi Santosa, and Diah Kristina. Ludic linguistics: A revisited taxonomy of fictional constructed language design approach for video games. *GEMA Online Journal of Language Studies*, 17(4), 2017.

[962] Doc Quigg. Machine writes TV plays, but virtue doesn't always win. *The Miami News*, page 3B, Oct 1960.

[963] Joe Quinn. TX-0 digital computer at MIT writes western drama to be seen on CBS-TV, 1960. Courtesy Computer History Museum.

[964] Racter. *The Policeman's Beard is Half Constructed*. 1984.

[965] Oz Radiano, Yonatan Graber, Moshe Mahler, Leonid Sigal, and Ariel Shamir. Story albums: Creating fictional stories from personal photograph sets. In *Computer Graphics Forum*, volume 37, pages 19–31. Wiley Online Library, 2018.

[966] Elahe Rahimtoroghi, Thomas Corcoran, Reid Swanson, Marilyn A Walker, Kenji Sagae, and Andrew Gordon. Minimal narrative annotation schemes and their applications. In *Proc. Intelligent Narrative Technologies*, 2014.

[967] Elahe Rahimtoroghi, Reid Swanson, Marilyn A Walker, and Thomas Corcoran. Evaluation, orientation, and action in interactive storytelling. *Proc. Intelligent Narrative Technologies*, 2013.

[968] Elahe Rahimtoroghi, Jiaqi Wu, Ruimin Wang, Pranav Anand, and Marilyn Walker. Modelling protagonist goals and desires in first-person narrative. In *Proc. Special Interest Group on Discourse and Dialogue*, pages 360–369, 2017.

[969] Shayndi Raice. Forget the Midwest. Minnesota casts itself as the North. *Wall Street Journal*, 2018.

[970] Ashwin Ram. Are you up to the challenge? announcing the Alexa Prize: $2.5 million to advance conversational artificial intelligence. `https://developer.amazon.com/blogs/post/Tx221UQAWNUXON3/Are-you-up-to-the-Challenge-Announcing-the-Alexa-Prize-2-5-Million-to-Advan` Last accessed: June 24, 2018.

[971] Ashwin Ram. Question-driven understanding: An integrated theory of story understanding, memory and learning. 1989.

[972] Ashwin Ram. Knowledge goals: A theory of interestingness. In *Proc. Cognitive Science Society*, pages 206–214, 1990.

[973] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. Conversational ai: The science behind the alexa prize. In *Proc. Alexa Prize 2017*, 2017.

[974] Richard Ramchurn. Brain controlled cinema. In *Proc. International BCS Human Computer Interaction Conference: Fusion!* BCS Learning & Development Ltd., 2016.

[975] Richard Ramchurn, Max L Wilson, Sarah Martindale, and Steve Benford. #scanners 2—the moment: A new brain-controlled movie. In *Proc. ACM*

CHI Conference Extended Abstracts on Human Factors in Computing Systems, page D210. ACM, 2018.

[976] Alejandro Ramirez and Vadim Bulitko. Automated planning and player modeling for interactive storytelling. *Transactions on Computational Intelligence and AI in Games*, 7(4):375–386, 2015.

[977] Stefan Rank, Steve Hoffmann, Hans-Georg Struck, Ulrike Spierling, and Paolo Petta. Creativity in configuring affective agents for interactive storytelling. In *Proc. International Conference on Computational Creativity*, pages 165–169, 2012.

[978] Jonathan Rappner. Dynamisk kunskapsspridning mellan spelkaraktärer genom konversationer [dynamic spread of knowledge among non-playing characters through conversations]. Bachelor's thesis, Malmö University, 2015.

[979] Philip Rawson. *Sculpture*. University of Pennsylvania Press, 1997.

[980] Amelia Rayno. How Eric Dayton convinced Minnesotans—and the rest of the country—to embrace the 'North'. *Star Tribune*, 2017.

[981] Aaron Reed. *18 Cadence*. 2006.

[982] Aaron A Reed. A sequence of possibilities. Master's thesis, University of California, Santa Cruz, 2011.

[983] Aaron A Reed. Sharing authoring with algorithms. In *Proc. Procedural Content Generation in Games*, 2012.

[984] Aaron A Reed. *Changeful Tales: Design-Driven Approaches Toward More Expressive Storygames*. PhD thesis, University of California, Santa Cruz, 2017.

[985] Aaron A Reed, Jacob Garbe, Noah Wardrip-Fruin, and Michael Mateas. Ice-Bound: Combining richly-realized story with expressive gameplay. In *Proc. International Conference on the Foundations of Digital Games*, 2014.

[986] Aaron A Reed, Ben Samuel, Anne Sullivan, Ricky Grant, April Grow, Justin Lazaro, Jennifer Mahal, Sri Kurniawan, Marilyn Walker, and Noah Wardrip-Fruin. Spyfeet: An exercise RPG. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 310–312. ACM, 2011.

[987] Aaron A Reed, Ben Samuel, Anne Sullivan, Ricky Grant, April Grow, Justin Lazaro, Jennifer Mahal, Sri Kurniawan, Marilyn A Walker, and Noah Wardrip-Fruin. A step towards the future of role-playing games: The spyfeet mobile rpg project. In *AIIDE*, 2011.

[988] Lena Reed, Jiaqi Wu, Shereen Oraby, Pranav Anand, and Marilyn Walker. Learning lexico-functional patterns for first-person affect. In *Proc. Association for Computational Linguistics*, volume 2, pages 141–147, 2017.

[989] Matthias Rehm and Elisabeth André. Catch me if you can: Exploring lying agents in social settings. In *Proc. International Conference on Autonomous Agents and Multiagent Systems*, pages 937–944. ACM, 2005.

[990] Jasia Reichardt. *Robots: Fact, fiction, and prediction*. Penguin Books, 1978.

[991] W Scott Reilly. The art of creating emotional and robust interactive characters. In *Working Notes of the AAAI Spring Symposium on Interactive Story Systems*, 1995.

[992] W Scott Reilly and Joseph Bates. Building emotional agents. Technical Report CM UC S-92-143, Carnegie Mellon University, 1992.

[993] Andrew Reinhard. Video games as archaeological sites: Treating digital entertainment as built environments. In A A Mol, C E Ariese-Vandemeulebroucke, K H J Boom, and A Politopoulos, editors, *The Interactive Past: Archaeology, Heritage, and Video Games*. Sidestone Press.

[994] Andrew Reinhard. *Archaeogaming: An introduction to archaeology in and of video games*. Berghahn Books, 2018.

[995] Henrique Daniel Santarém Reis. Lie to me: Lying virtual agents. Master's thesis, Universidade Técnica de Lisboa, 2012.

[996] J Kirkland Reynolds and Kathy Pezdek. Face recognition memory: The effects of exposure duration and encoding instruction. *Applied Cognitive Psychology*, 6(4):279–292, 1992.

[997] Brad Rhodes and Pattie Maes. The stage as a character: Automatic creation of acts of god for dramatic effect. In *Working Notes of the AAAI Spring Symposium on Interactive Story Systems*, 1995.

[998] Martin Rhodes, Simon Coupland, and Tracy Cruickshank. Enhancing real-time sports commentary generation with dramatic narrative devices. In *Proc. Joint International Conference on Interactive Digital Storytelling*, pages 111–116. Springer, 2010.

[999] Richard H Richens. Preprogramming for mechanical translation. *Mechanical Translation*, 3(1):20–25, 1956.

[1000] Mark O Riedl and Andrew Stern. Believable agents and intelligent story adaptation for interactive storytelling. In *Proc. International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pages 1–12. Springer, 2006.

[1001] Mark O Riedl and Andrew Stern. Failing believably: Toward drama management with autonomous actors in interactive narratives. In *Proc. International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pages 195–206. Springer, 2006.

[1002] Mark O Riedl and R Michael Young. Character-focused narrative generation for execution in virtual worlds. In *Proc. International Conference on Virtual Storytelling*, pages 47–56. Springer, 2003.

[1003] Mark O Riedl and Robert Michael Young. From linear story generation to branching story graphs. *Computer Graphics and Applications*, 26(3):23–31, 2006.

[1004] Mark O Riedl and Robert Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.

[1005] Mark Owen Riedl. *Narrative generation: balancing plot and character*. PhD thesis, North Carolina State University, 2004.

[1006] Mark Owen Riedl and R Michael Young. An intent-driven planner for multi-agent story generation. In *Proc. International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 186–193. IEEE, 2004.

[1007] Christopher K Riesbeck and Roger C Schank. *Inside case-based reasoning*. Psychology Press, 2013.

[1008] Elena Rishes, Stephanie M Lukin, David K Elson, and Marilyn A Walker. Generating different story tellings from semantic representations of narrative. In *Proc. International Conference on Interactive Digital Storytelling*, pages 192–204. Springer, 2013.

[1009] Graeme Ritchie. A rational reconstruction of the Proteus sentence planner. In *Proc. International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics*, pages 327–329. Association for Computational Linguistics, 1984.

[1010] David L Roberts and Charles L Isbell. A survey and qualitative analysis of recent advances in drama management. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, 4(2):61–75, 2008.

[1011] Justus Robertson, Adam Amos-Binks, and R. Michael Young. Directing intentional superposition manipulation. In *Proc. Intelligent Narrative Technologies*, 2017.

[1012] Justus Robertson and R Michael Young. A model of superposed states. In *Proc. Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[1013] Justus Robertson and R Michael Young. Perceptual experience management. *Transactions on Games*, 2018.

[1014] Justus Robertson and Robert Michael Young. Finding Schrödinger's gun. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2014.

[1015] Richard W Robins, R Chris Fraley, Brent W Roberts, and Kali H Trzesniewski. A longitudinal study of personality change in young adulthood. *Journal of Personality*, 69(4):617–640, 2001.

[1016] Douglas LT Rohde, Steve Olson, and Joseph T Chang. Modelling the recent common ancestry of all living humans. *Nature*, 431(7008):562, 2004.

[1017] Jason Rohrer. *Passage*. 2007.

[1018] Jason Rohrer. What i was trying to do with Passage. `http://hcsoftware.sourceforge.net/passage/statement.html`, November 2007.

[1019] Jason Rohrer. *One Hour One Life*. 2018.

[1020] Miguel Roig-Francolí. *Understanding post-tonal music*. McGraw-Hill, 2008.

[1021] Hanna-Riikka Roine. *Imaginative, Immersive and Interactive Engagements: The Rhetoric of Worldbuilding in Contemporary Speculative Fiction*. Tampere University Press, 2016.

[1022] Ruth Ronen. *Possible worlds in literary theory*, volume 7. Cambridge University Press, 1994.

[1023] Richard Rorty. Philosophy and the mirror of nature. 1979.

[1024] Richard Rorty. Solidarity or objectivity? In Michael Krausz, editor, *Relativism: Interpretation and Confrontation*, pages 167–183. University of Notre Dame Press, 1989.

[1025] Richard McKay Rorty, Richard Rorty, et al. *Objectivity, relativism, and truth: philosophical papers*, volume 1. Cambridge University Press, 1991.

[1026] Montague Henrietta Rosa. Name selector, Aug 1947. US Patent 2,425,197.

[1027] Mike Rose. Dwarf Fortress in 2013. `http://www.gamasutra.com/view/feature/195148/dwarf_fortress_in_2013.php`, Jul 2018.

[1028] Douglas T Ross. Note taken by douglas t. ross, 1960. Douglas T. Ross papers. MC0414, box 217. Institute Archives and Special Collections, MIT Libraries, Cambridge, Massachusetts.

[1029] Douglas T Ross. Origins of the APT language for automatically programmed tools. In *Proc. History of Programming Languages I*, pages 279–338. ACM, 1978.

[1030] Douglas Taylor Ross. Computer-aided design: A statement of objectives. Technical report, MIT Electronic Systems Laboratory, 1960.

[1031] Peter Ross. Postgraduate prospectus for entry in 1998. Technical report, University of Edinburgh, 1997.

[1032] Marcos Ruiz-Soler and Francesc S Beltran. The relative salience of facial features when differentiating faces based on an interference paradigm. *Journal of Nonverbal Behavior*, 36(3):191–203, 2012.

[1033] David E Rumelhardt. Understanding and summarizing brief stories. Technical Report Center for Human Information Processing Tech. Report No. 58, University of California, San Diego, 1976.

[1034] David E Rumelhart. Notes on a schema for stories. In *Representation and understanding*, pages 211–236. 1975.

[1035] Bertrand Russell. On denoting. *Mind*, 14(56):479–493, 1905.

[1036] Stuart J Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education Ltd., 2016.

[1037] James Ryan. Diegetically grounded evolution of gameworld languages. In *Proc. Procedural Content Generation*, 2016.

[1038] James Ryan. *Situated and Pragmatic Story Generation*. Unpublished thesis proposal, 2016.

[1039] James Ryan. Grimes' fairy tales: a 1960s story generator. In *Proc. International Conference on Interactive Digital Storytelling*, pages 89–103. Springer, 2017.

[1040] James Ryan. *Curating Simulated Storyworlds*. PhD thesis, University of California, Santa Cruz, 2018.

[1041] James Ryan. A garden, a forking path: Interactive branching narrative in The Lady of May (1578). In *Proc. History of Expressive Systems*, to appear.

[1042] James Ryan, Tyler Brothers, Michael Mateas, and Noah Wardrip-Fruin. Juke Joint: Characters who are moved by music. *Proc. Experimental AI in Games*, 2016.

[1043] James Ryan, Michael Matea s, and Noah Wardrip-Fruin. A simple method for evolving large character social networks. In *Proc. Social Believability in Games*, 2016.

[1044] James Ryan and Michael Mateas. Simulating character knowledge phenomena in Talk of the Town. In Steve Rabin, editor, *Game AI Pro 3: Collected Wisdom of Game AI Professionals*, pages 433–448. CRC Press, 2017.

[1045] James Ryan, Michael Mateas, and Noah Wardrip-Fruin. Characters who speak their minds: Dialogue generation in Talk of the Town. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2016.

[1046] James Ryan, Michael Mateas, and Noah Wardrip-Fruin. Generative character conversations for background believability and storytelling. *Proc. Social Believability in Games*, 2016.

[1047] James Ryan and Mark J Nelson. 1st Workshop on the History of Expressive Systems. In *Proc. International Conference on Interactive Digital Storytelling*, pages 409–412. Springer, 2017.

[1048] James Ryan, Ben Samuel, and Adam Summerville. Bad News: Behind the curtain. Technical report, 2017.

[1049] James Ryan, Ethan Seither, Michael Mateas, and Noah Wardrip-Fruin. Expressionist: An authoring tool for in-game text generation. In *Proc. International Conference on Interactive Digital Storytelling*, 2016.

[1050] James Ryan, Adam James Summerville, Michael Mateas, and Noah Wardrip-Fruin. Translating player dialogue into meaning representations using LSTMs. In *Proc. International Conference on Intelligent Virtual Agents*, pages 383–386. Springer, 2016.

[1051] James Ryan and Reid Swanson. Recognizing coherent narrative blog content. In *Proc. International Conference on Interactive Digital Storytelling*, pages 234–246. Springer, 2016.

[1052] James Ryan, Marilyn A Walker, and Noah Wardrip-Fruin. Toward recombinant dialogue in interactive narrative. In *Proc. Intelligent Narrative Technologies*, 2014.

[1053] James Owen Ryan, Casey Barackman, Nicholas Kontje, Taylor Owen-Milner, Marilyn A Walker, Michael Mateas, and Noah Wardrip-Fruin. Combinatorial dialogue authoring. In *Proc. International Conference on Interactive Digital Storytelling*, pages 13–24. Springer, 2014.

[1054] James Owen Ryan, Andrew Max Fisher, Taylor Owen-Milner, Michael Mateas, and Noah Wardrip-Fruin. Toward natural language generation by humans. *Proc. Intelligent Narrative Technologies and Social Believability in Games*, 2015.

[1055] James Owen Ryan, Eric Kaltman, Andrew Max Fisher, Timothy Hong, Taylor Owen-Milner, Michael Mateas, and Noah Wardrip-Fruin. Large-scale interactive visualizations of nearly 12,000 digital games. In *Proc. International Conference on the Foundations of Digital Games*, 2015.

[1056] James Owen Ryan, Eric Kaltman, Timothy Hong, Michael Mateas, and Noah Wardrip-Fruin. People tend to like related games. In *Proc. International Conference on the Foundations of Digital Games*, 2015.

[1057] James Owen Ryan, Eric Kaltman, Michael Mateas, and Noah Wardrip-Fruin. What we talk about when we talk about games: Bottom-up game studies using natural language processing. In *Proc. International Conference on the Foundations of Digital Games*, 2015.

[1058] James Owen Ryan, Michael Mateas, and Noah Wardrip-Fruin. Open design challenges for interactive emergent narrative. In *Proc. International Conference on Interactive Digital Storytelling*, pages 14–26. Springer, 2015.

[1059] James Owen Ryan, Michael Mateas, and Noah Wardrip-Fruin. A lightweight videogame dialogue manager. In *Proc. International Joint Conference of DiGRA and FDG*, 2016.

[1060] James Owen Ryan, Ben Samuel, Adam Summerville, and Jonathan Lessard. Bad News: A computationally assisted live-action prototype to guide content creation. *Proc. Experimental AI in Games*, 2015.

[1061] James Owen Ryan, Adam Summerville, Michael Mateas, and Noah Wardrip-Fruin. Toward characters who observe, tell, misremember, and lie. In *Proc. Experimental AI in Games*, pages 56–62, 2015.

[1062] James Owen Ryan, Adam J Summerville, and Ben Samuel. Bad News: A game of death and communication. In *Proc. CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 160–163. ACM, 2016.

[1063] Marie-Laure Ryan. Embedded narratives and tellability. *Style*, pages 319–340, 1986.

[1064] Marie-Laure Ryan. Embedded narratives and the structure of plans. *Text: Interdisciplinary Journal for the Study of Discourse*, 6(1):107–141, 1986.

[1065] Marie-Laure Ryan. The heuristics of automatic story generation. *Poetics*, 16(6):505–534, 1987.

[1066] Marie-Laure Ryan. *Possible worlds, artificial intelligence, and narrative theory*. Indiana University Press, 1991.

[1067] Marie-Laure Ryan. Interactive drama: Narrativity in a highly interactive environment. *Modern Fiction Studies*, 43(3):677–707, 1997.

[1068] Marie-Laure Ryan. Postmodernism and the doctrine of panfictionality. *Narrative*, 5(2):165–187, 1997.

[1069] Marie-Laure Ryan. *Narrative as virtual reality: Immersion and interactivity in literature and electronic media*. Johns Hopkins University Press, 2001.

[1070] Marie-Laure Ryan. *Avatars of story*. University of Minnesota Press, 2006.

[1071] Marie-Laure Ryan. From narrative games to playable stories: Toward a poetics of interactive narrative. *Storyworlds: A Journal of Narrative Studies*, 1(1):43–59, 2009.

[1072] Marie-Laure Ryan. Narration in various media. *Handbook of Narratology*, pages 468–88, 2015.

[1073] Warren Sack. Micro-Talespin: A story generator. `http://eliterature.org/images/microtalespin.txt`, 1992. Last accessed: May 25, 2018.

[1074] Warren Sack. HOXs: Histories of software systems, textbooks, translations and miniatures. In *Proc. History of Expressive Systems*. In press.

[1075] Warren Sack and Sophie Bargues Rollins. Une machine à raconter des histoires: Propp et les software studies. *Les Temps Modernes*, (5):216–243, 2013. English translation obtained from author.

[1076] Harvey Sacks. *Lectures on conversation: Volumes I & II*. Blackwell, 1992.

[1077] Kenji Sagae, Andrew S Gordon, Morteza Dehghani, Mike Metke, Jackie S Kim, Sarah I Gimbel, Christine Tipper, Jonas Kaplan, and Mary Helen Immordino-Yang. A data-driven approach for classification of subjectivity in personal narratives. In *OpenAccess Series in Informatics*, volume 32. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[1078] Neelima Sailaja, Andy Crabtree, and Phil Stenton. Challenges of using personal data to drive personalised electronic programme guides. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5226–5231. ACM, 2017.

[1079] Katie Salen and Eric Zimmerman. *Rules of play: Game design fundamentals*. MIT Press, 2004.

[1080] Anastasia Salter. *What is your quest?: From adventure games to interactive books*. University of Iowa Press, 2014.

[1081] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2):187–260, 1984.

[1082] Mark Sample. An account of randomness in literary computing. `http://www.samplereality.com/2013/01/08/an-account-of-randomness-in-literary-computing/`, 2013. Presented at Modern Language Association.

[1083] Walker Sampson. Interview with Tarn Adams [developer/designer/founder series]. `http://hdl.handle.net/2152/8596`, 2009.

[1084] Ben Samuel. *Crafting Stories Through Play*. PhD thesis, University of California, Santa Cruz, 2016.

[1085] Ben Samuel, Dylan Lederle-Ensign, and Mike Treanor. Playing the worlds of Prom Week. In *Narrative Theory, Literature, and New Media*, pages 99–117. Routledge, 2015.

[1086] Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. The design of Writing Buddy: a mixed-initiative approach towards computational story collaboration. In *Proc. International Conference on Interactive Digital Storytelling*, pages 388–396. Springer, 2016.

[1087] Ben Samuel, Josh McCoy, Mike Treanor, Aaron A Reed, Michael Mateas, and Noah Wardrip-Fruin. Introducing story sampling: Preliminary results of a new interactive narrative evaluation technique. In *Proc. International Conference on the Foundations of Digital Games*, 2014.

[1088] Ben Samuel, Aaron Reed, Emily Short, Samantha Heck, Barrie Robison, Landon Wright, Terence Soule, Mike Treanor, Joshua McCoy, Anne Sullivan, Alireza Shirvani, Edward Garcia, Rachelyn Farrell, Stephen Ware, and Katherine Compton. Playable experiences at AIIDE 2018. pages 275–280, 2018.

[1089] Ben Samuel, Aaron A Reed, Paul Maddaloni, Michael Mateas, and Noah Wardrip-Fruin. The Ensemble engine: Next-generation social physics. In *Proc. International Conference on the Foundations of Digital Games*, pages 22–25, 2015.

[1090] Ben Samuel, James Ryan, Adam Summerville, Michael Mateas, and Noah Wardrip-Fruin. Bad News: An experiment in computationally assisted performance. In *Proc. International Conference on Interactive Digital Storytelling*, pages 108–120, 2016.

[1091] Ben Samuel, James Ryan, Adam Summerville, Michael Mateas, and Noah Wardrip-Fruin. Computatrum personae: toward a role-based taxonomy of (computationally assisted) performance. In *Proc. Experimental AI in Games*, pages 79–85, 2016.

[1092] Edward Sapir. Language: An introduction to the study of speech. 1921.

[1093] Theodore R Sarbin, editor. *Narrative psychology: The storied nature of human conduct*. Praeger Publishers/Greenwood Publishing Group, 1986.

[1094] Theordore R Sarbin. The narrative as a root metaphor for psychology. In Theordore R Sarbin, editor, *Narrative psychology: The storied nature of human conduct*. Praeger Publishers/Greenwood Publishing Group, 1986.

[1095] CJ Saretto and R Michael Young. Mediation in mimesis liquid narratives. In *Proc. ACM Southeast Conference*, pages 120–126, 2001.

[1096] Marcin Sarnek. Kings don't die in traffic accidents. *Cryptohistories*, 2015.

[1097] Theophilus Savvas. 'there is only narrative': El doctorow and postmodernism. In *American Postmodernist Fiction and the Past*, pages 124–155. Springer, 2011.

[1098] Jean-Marie Schaeffer. Fiction, pretense, and narration. *Style*, 32, 1998.

[1099] Jean-Marie Schaeffer. Fictional vs. factual narration. *Handbook of Narratology*, 19:98–114, 2009.

[1100] Jean-Marie Schaeffer. *Why Fiction?* University of Nebraska Press, 2010.

[1101] Roger C Schank. The primitive ACTs of conceptual dependency. In *Theoretical Issues in Natural Language Processing*, 1975.

[1102] Roger C Schank. Inference in the conceptual dependency paradigm: A personal history. Technical Report 141, Yale University, 1978.

[1103] Roger C Schank. Interestingness: controlling inferences. *Artificial Intelligence*, 12(3):273–297, 1979.

[1104] Roger C Schank. *Dynamic memory: A theory of reminding and learning in computers and people*, volume 240. Cambridge University Press, 1982.

[1105] Roger C Schank and Robert P Abelson. Scripts, plans, and knowledge. In *Proc. International Joint Conference on Artificial Intelligence*, pages 151–157, 1975.

[1106] Roger C Schank, Janet L Kolodner, and Gerald DeJong. Conceptual information retrieval. In *Proc. Research and development in information retrieval*, pages 94–116. Association for Computing Machinery, 1980.

[1107] Roger C Schank and Christopher K Riesbeck. *Inside computer understanding: Five programs plus miniatures*. Routledge, 1981.

[1108] Michael Scheffel. Narrative constitution. In Peter Hühn et al., editors, *Living Handbook of Narratology*. Hamburg University Press, 2010.

[1109] Christian Scheier and Dimitrios Lambrinos. Categorization in a real-world agent using haptic exploration and active perception. In *From animals to animats: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 65–75. MIT Press, 1996.

[1110] Thomas C Schelling. Dynamic models of segregation. *Journal of mathematical sociology*, 1(2):143–186, 1971.

[1111] Klaus R Scherer, Angela Schorr, and Tom Johnstone. *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, 2001.

[1112] Deborah Schiffrin. *Discourse markers*. Number 5. Cambridge University Press, 1988.

[1113] Wolf Schmid. *Die narrativen Ebenen 'Geschehen', 'Geschichte','Erzählung' und 'Präsentation der Erzählung'*, volume 9. Wiener Slawistischer Almanach, 1982.

[1114] Wolf Schmid. Narrativity and eventfulness. *What is Narratology*, pages 17–33, 2003.

[1115] Wolf Schmid. *Narratology: an introduction.* Walter de Gruyter, 2010.

[1116] David P Schmitt, Jüri Allik, Robert R McCrae, and Verónica Benet-Martínez. The geographic distribution of big five personality traits: Patterns and profiles of human self-description across 56 nations. *Journal of Cross-Cultural Psychology*, 38(2):173–212, 2007.

[1117] Brad Schultz. Publisher claims computer composed novel. *Computerworld*, page 23, Aug 1980.

[1118] Katharine Schwab. This ai paints like the old masters. can you tell the difference? `https://www.fastcodesign.com/90167584/this-ai-paints-like-the-old-masters-can-you-tell-the-difference`, Apr 2018.

[1119] Helen J Schwartz. The student as producer and consumer of text: Computer uses in English studies. *Journal of Teaching Writing*, 7(1):57–66, 1988.

[1120] William Seager. Panpsychism. *Encyclopedia of Cognitive Science*, 2005.

[1121] William Seager. Emergence and cellular automata. In *Natural Fabrications*, pages 65–83. Springer, 2012.

[1122] John R Searle. The logical status of fictional discourse. *New Literary History*, 6(2):319–332, 1975.

[1123] John R Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3):417–424, 1980.

[1124] John R Searle. *The rediscovery of the mind.* MIT Press, 1992.

[1125] Kelton Sears. How two brothers from Silverdale made the worldâĂŹs most complex video game. *Seattle Weekly*, Jun 2016.

[1126] Maarten Selfhout, William Burk, Susan Branje, Jaap Denissen, Marcel Van Aken, and Wim Meeus. Emerging late adolescent friendship networks and big five personality traits: A social network approach. *Journal of Personality*, 78(2):509–538, 2010.

[1127] Phoebe Sengers. Do the thing right: an architecture for action-expression. In *Proc. International Conference on Autonomous Agents*, pages 24–31. ACM, 1998.

[1128] Michael J Sennott. Quicksilver: Infinite story procedurally generated episodic narratives for gameplay. Master's thesis, University of Southern California, 2012.

[1129] Amazon Web Services. Amazon Polly. `https://aws.amazon.com/polly/`. Last accessed: June 24, 2018.

[1130] Nikitas M Sgouros and Sophia Kousidou. Generation and implementation of mixed-reality, narrative performances involving robotic actors. In *International Conference on Virtual Storytelling*, pages 69–78. Springer, 2001.

[1131] Nikitas M Sgouros, Sophia Kousidou, and Yiannis Melanitis. Terra Incognita: A multimedia and robotics interactive performance centered around edgar allan poe's" dream-land. *Interactive Multimedia Electronic J. Computer-Enhanced Learning,*, 4(1), 2002.

[1132] Noor Shaker, Antonios Liapis, Julian Togelius, Ricardo Lopes, and Rafael Bidarra. Constructive generation methods for dungeons and levels. In Noor Shaker, Julian Togelius, and Mark J Nelson, editors, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, pages 31–55. Springer, 2016.

[1133] Noor Shaker, Gillian Smith, and Georgios N Yannakakis. Evaluating content generators. In *Procedural Content Generation in Games*, pages 215–224. Springer, 2016.

[1134] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2015.

[1135] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[1136] William Shakespeare. *Macbeth*. University Press, 1901.

[1137] Claude Elwood Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[1138] Daniel Shapiro, Larry LeBron, Andrew Stern, and Michael Mateas. Aesthetic interleaving of character performance requests. In *Proc. Intelligent Narrative Technologies and Social Believability in Games*, 2015.

[1139] Daniel Shapiro, Karen Tanenbaum, Josh McCoy, Larry LeBron, Craig Reynolds, Andrew Stern, Michael Mateas, Bill Ferguson, David Diller, Kerry Moffitt, Will Coon, and Bruce Roberts. Composing social interactions via social games. In *Proc. International Conference on Autonomous Agents and Multiagent Systems*, pages 573–580, 2015.

[1140] Daniel G Shapiro, Joshua McCoy, April Grow, Ben Samuel, Andrew Stern, Reid Swanson, Mike Treanor, and Michael Mateas. Creating playable social experiences through whole-body interaction with virtual characters. In *AIIDE*, 2013.

[1141] Michael Gordon Shapiro. The Penguin Who Wouldn't Swim. `http://mikemusic.com/album/zoesisstudios/`. Last accessed: May 25, 2018.

[1142] Stuart C Shapiro. The SNePS semantic network processing system. In *Associative Networks*, pages 179–203. Elsevier, 1979.

[1143] Andrew Sharp. *Ancient voyagers in Polynesia*. University of California Press, 1964.

[1144] Yotam Shibolet, Noam Knoller, and Hartmut Koenitz. A framework for classifying and describing authoring tools for interactive digital narrative. In *Proc. International Conference on Interactive Digital Storytelling*. Springer, 2018.

[1145] Eugénie Shinkle. Videogames and the digital sublime. In *Digital Cultures and the Politics of Emotion*, pages 94–108. Springer, 2012.

[1146] Alireza Shirvani, Rachelyn Farrell, and Stephen G Ware. Combining intentionality and belief: Revisiting believable character plans. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, pages 222–228, 2018.

[1147] Alireza Shirvani, Stephen G Ware, and Rachelyn Farrell. A possible worlds model of belief for state-space narrative planning. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, pages 101–107, 2017.

[1148] Ben Shneiderman. Creativity support tools. *Communications of the ACM*, 45(10):116–120, 2002.

[1149] Emily Short. Bowls of oatmeal and text generation. *Emily Short's Interactive Storytelling*, Sep 2016.

[1150] Tanya X. Short. Designing stronger ai personalities. In *Proc. Experimental AI in Games*, pages 111–117, 2017.

[1151] Patrick E Shrout and Susan T Fiske. *Personality research, methods, and theory: A festschrift honoring Donald W. Fiske*. Psychology Press, 2014.

[1152] Mei Si, Stacy C Marsella, and David V Pynadath. Thespian: Using multi-agent fitting to craft interactive drama. In *Proc. International Conference on Autonomous Agents and Multiagent Systems*, pages 21–28. ACM, 2005.

[1153] Miguel Sicart. Defining game mechanics. *Game Studies*, 8(2):1–14, 2008.

[1154] Miguel Sicart. Loops and metagames: Understanding game design structures. In *Proc. International Conference on the Foundations of Digital Games*, 2015.

[1155] Robert Simmons and Jonathan Slocum. Generating english discourse from semantic networks. *Communications of the ACM*, 15(10):891–905, 1972.

[1156] Robert F Simmons. The clowns microworld. In *Proc. Theoretical Issues in Natural Language Processing*, pages 17–19. Association for Computational Linguistics, 1975.

[1157] Robert F Simmons and Alfred Correira. Rule forms for verse, sentences, and story trees. In *Associative Networks*, pages 363–392. 1979.

[1158] Robert F Simmons, Sheldon Klein, and Keren McConlogue. Toward the synthesis of human language behavior. *Systems Research and Behavioral Science*, 7(3):402–407, 1962.

[1159] Robert F Simmons, Sheldon Klein, and Keren McConlogue. Indexing and dependency logic for answering English questions. *American Documentation*, 15(3):196–204, 1964.

[1160] Herbert A Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

[1161] Jan Simons et al. Narrative, games, and theory. *Game Studies*, 7(1):1–21, 2007.

[1162] Sims Wiki. Wants and fears. `http://sims.wikia.com/wiki/Wants_and_fears`. Last accessed: June 24, 2018.

[1163] James Skorupski, Lakshmi Jayapalan, Sheena Marquez, and Michael Mateas. Wide Ruled: A friendly interface to author-goal based story generation. In *Proc. International Conference on Virtual Storytelling*, pages 26–37. Springer, 2007.

[1164] James Skorupski, Josh McCoy, Catherine Zanbaka, Kathy Ryall, and Michael Mateas. Three avatars walk into a bar: Defining and evaluating realism for virtual supporting characters. In *Proc. International Conference on the Foundations of Digital Games*, 2012.

[1165] Stephen Slade. The Yale Artificial Intelligence Project: A brief history. *AI Magazine*, 8(4):67, 1987.

[1166] John Slaney and Sylvie Thiébaux. Blocks world revisited. *Artificial Intelligence*, 125(1-2):119–153, 2001.

[1167] Sarah Sloane. *Interactive fiction, virtual realities, and the reading-writing relationship*. PhD thesis, The Ohio State University, 1991.

[1168] Adam M Smith, Chris Lewis, Kenneth Hullett, Gillian Smith, and Anne Sullivan. An inclusive taxonomy of player modeling. Technical Report UCSC-SOE-11-13, University of California, Santa Cruz, 2011.

[1169] Brian Cantwell Smith. *Procedural reflection in programming languages*. PhD thesis, Massachusetts Institute of Technology, 1982.

[1170] Elliot Smith and Peter Hancox. Representation, coherence and inference. *Artificial Intelligence Review*, 15(4):295–323, 2001.

[1171] Gillian Smith. The future of procedural content generation in games. In *Proc. Experimental AI in Games*, pages 53–57, 2014.

[1172] Gillian Smith. An analog history of procedural content generation. In *Proc. Foundations of Digital Games*, 2015.

[1173] Gillian Smith. Procedural content generation: An overview. In Steve Rabin, editor, *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, pages 501–518. CRC Press, 2015.

[1174] Gillian Smith. What do we value in procedural content generation? In *Proc. International Conference on the Foundations of Digital Games*. ACM, 2017.

[1175] Gillian Smith. What do we value in procedural content generation? In *Proc. International Conference on the Foundations of Digital Games*, page 69. ACM, 2017.

[1176] Gillian Smith, Mirjam P Eladhari, Matthew J Guzdial, Emily Short, Adam M Smith, Anne Sullivan, Tommy Thompson, and R Michael Young. Ai as reflective practice. In *Artificial and Computational Intelligence in Games: AI-Driven Game Design*, pages 113–115. 2018.

[1177] Gillian Smith and Jim Whitehead. Analyzing the expressive range of a level generator. In *Proc. Procedural Content Generation in Games*, page 4. ACM, 2010.

[1178] L C Smith. Recall of superordinate and subordinate concepts from science and social studies text. In *Proc. American Educational Research Association*, 1985.

[1179] Sean Smith and Joseph Bates. Towards a theory of narrative for interactive fiction. Technical Report CMU-CS-89-121, Carnegie Mellon University, 1989.

[1180] Tony C Smith and Ian H Witten. A planning mechanism for generating story text. *Literary and Linguistic Computing*, 6(2):119–126, 1991.

[1181] William Jay Smith. *The World Below the Window: Poems 1937–1997*. Johns Hopkins University Press, 2001.

[1182] Stephen Soldz and George E Vaillant. The Big Five personality traits and the life course: A 45-year longitudinal study. *Journal of Research in Personality*, 33(2):208–232, 1999.

[1183] Richard Solomon. Archives: The Computer Pioneers: The TX-0. `http://ethw.org/Archives:The_Computer_Pioneers:_The_TX-0`, 1983.

[1184] John F Sowa. Semantic networks. *Encyclopedia of artificial intelligence*, 2:1493–1511, 1992.

[1185] Dietrich Squinkifer and Jess Marcotte. Most Sincere Greetings, Esteemed One. `http://squinky.me/2016/01/31/most-sincere-greetings-esteemed-one/`, 2016.

[1186] Dietrich Squinkifer and Jess Marcotte. rustle your leaves to me softly. `http://squinky.me/2017/01/28/rustle-your-leaves-to-me-softly/`, 2017.

[1187] Dietrich Squinkifer and Jess Marcotte. The Truly Terrific Traveling Troubleshooter. `http://squinky.me/2017/01/28/rustle-your-leaves-to-me-softly/`, 2017.

[1188] Dietrich Squinkifer and Jess Marcotte. The Truly Terrific Traveling Troubleshooter. `https://handsomefoxes.wordpress.com/`, 2017.

[1189] Dietrich Squinkifer (writing as Deirdra Kiai). Coffee: A Misunderstanding. `https://escholarship.org/uc/item/9k43n54m`, 2014.

[1190] Sanjay Srivastava, Oliver P John, Samuel D Gosling, and Jeff Potter. Development of personality in early and middle adulthood: Set like plaster or persistent change? *Journal of Personality and Social Psychology*, 84(5):1041, 2003.

[1191] J Frits Staal. Context-sensitive rules in pāṇini. *Foundations of Language*, pages 63–72, 1965.

[1192] Luc Steels. The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1–34, 1997.

[1193] Luc Steels. Modeling the cultural evolution of language. *Physics of Life Reviews*, 8(4):339–356, 2011.

[1194] Abraham Daniel Stein. Televisual sports videogames. Master's thesis, Massachusetts Institute of Technology, 2013.

[1195] Karl E Steiner and Thomas G Moher. Graphic StoryWriter: An interactive environment for emergent storytelling. In *Proc. SIGCHI conference on Human Factors in Computing Systems*, pages 357–364. ACM, 1992.

[1196] Tim Stephens. UC Santa Cruz creates new Department of Computational Media. `https://news.ucsc.edu/2014/10/computational-media.html`, Oct 2014.

[1197] Tim Stephens. UC Santa Cruz now offering graduate degrees in computational media. `https://news.ucsc.edu/2016/11/computational-media.html`, Nov 2016.

[1198] Andrew Stern. AI beyond computer games. In *Proc. AAAI Spring Symposium on Artificial Intelligence and Computer Games*, 1999.

[1199] Andrew Stern. Virtual babyz: Believable agents with narrative intelligence. Working Notes of the 1999 AAAI Spring Symposium on Narrative Intelligence, AAAI Press, 1999.

[1200] Andrew Stern. Embracing the combinatorial explosion: A brief prescription for interactive story r&d. In *Proc. International Conference on Interactive Digital Storytelling*, pages 1–5. Springer, 2008.

[1201] Andrew Stern, Adam Frank, and Ben Resner. Virtual Petz (video session): a hybrid approach to creating autonomous, lifelike dogz and catz. In *Proc. Second International Conference on Autonomous Agents*, pages 334–335. ACM, 1998.

[1202] Meir Sternberg. *The poetics of biblical narrative: Ideological literature and the drama of reading.* Indiana University Press, 1987.

[1203] Hito Steyerl. A sea of data: Apophenia and pattern (mis-)recognition. *E-Flux*, 72, Apr 2016.

[1204] Karlheinz Stierle. Geschehen, geschichte, text der geschichte. pages 530–534. Geschichte-Ereignis und Erzählung, 1973 [1971].

[1205] Christopher Strachey. The 'thinking' machine. *Encounter*, 3(4):25–31, 1954.

[1206] Alina Striner, Sasha Azad, and Chris Martens. Characterizing a spectrum of audience interactivity. Technical Report 1710.03320 (preprint), arXiv, 2017.

[1207] The Sims Studio. *The Sims 3*. Electronic Arts, 2009.

[1208] Bethesda Game Studios. *The Elder Scrolls V: Skyrim*. Bethesda Softworks, 2015.

[1209] Ludeon Studios. *RimWorld*. Ludeon Studios, 2013.

[1210] Anne Sullivan, April Grow, Tabitha Chirrick, Max Stokols, Noah Wardrip-Fruin, and Michael Mateas. Extending CRPGs as an interactive storytelling form. In *Proc. International Conference on Interactive Digital Storytelling*, pages 164–169. Springer, 2011.

[1211] Anne Sullivan, April Grow, Michael Mateas, and Noah Wardrip-Fruin. The design of Mismanor: creating a playable quest-based story game. In *Proc. International Conference on the Foundations of Digital Games*, pages 180–187. ACM, 2012.

[1212] Anne Margaret Sullivan. *The Grail Framework: Making stories playable on three levels in CRPGs*. PhD thesis, University of California, Santa Cruz, 2012.

[1213] Adam Summerville. *Learning from Games for Generative Purposes*. PhD thesis, University of California, Santa Cruz, 2018.

[1214] Adam Summerville, Morteza Behrooz, Michael Mateas, and Arnav Jhala. What does that ?-block do? learning latent causal affordances from Mario play traces. In *Proc. AAAI Workshop on What's Next for AI in Games*, volume 2017, 2017.

[1215] Adam Summerville, Julian RH Mariño, Sam Snodgrass, Santiago Ontañón, and Levi HS Lelis. Understanding Mario: an evaluation of design metrics for platformers. In *Proc. International Conference on the Foundations of Digital Games*, page 8. ACM, 2017.

[1216] Adam Summerville, Chris Martens, Sarah Harmon, Michael Mateas, Joseph Carter Osborn, Noah Wardrip-Fruin, and Arnav Jhala. From mechanics to meaning. *Transactions on Computational Intelligence and AI in Games*, 2017.

[1217] Adam Summerville, Chris Martens, Ben Samuel, Joseph Carter Osborn, Noah Wardrip-Fruin, and Michael Mateas. Gemini: Bidirectional generation and analysis of games via asp. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2018.

[1218] Adam Summerville and Michael Mateas. Super Mario as a string: Platformer level generation via LSTMs. In *Proc. International Joint Conference of DiGRA and FDG*, 2016.

[1219] Adam James Summerville, Shweta Philip, and Michael Mateas. MCMCTS PCG 4 SMB: Monte Carlo tree search to guide platformer level generation. In *Proc. Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[1220] Adam James Summerville, James Ryan, Michael Mateas, and Noah Wardrip-Fruin. CFGs-2-NLU: Sequence-to-sequence learning for mapping utterances to semantics and pragmatics. Technical Report UCSC-SOE-16-11, University of California, Santa Cruz, 2016.

[1221] Frederick Suppe. *The structure of scientific theories*, volume 634. University of Illinois Press, 1977.

[1222] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[1223] Neil Suttie, Sandy Louchart, Ruth Aylett, and Theodore Lim. Theoretical considerations towards authoring emergent narrative. In *Proc. International Conference on Interactive Digital Storytelling*, pages 205–216. Springer, 2013.

[1224] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT Press, 2018.

[1225] Reid Swanson. *Enabling open domain interactive storytelling using a data-driven case-based approach.* PhD thesis, University of Southern California, 2010.

[1226] Reid Swanson and Andrew S Gordon. Say Anything: A massively collaborative open domain story writing companion. In *Proc. International Conference on Interactive Digital Storytelling*, pages 32–40. Springer, 2008.

[1227] Reid Swanson and Andrew S Gordon. Say Anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *Transactions on Interactive Intelligent Systems*, 2(3):16, 2012.

[1228] Reid Swanson, Elahe Rahimtoroghi, Thomas Corcoran, and Marilyn Walker. Identifying narrative clause types in personal stories. In *Proc. Special Interest Group on Discourse and Dialogue*, pages 171–180, 2014.

[1229] IMT Swartjes and Mariët Theune. Late commitment: virtual story characters that can frame their world. Technical report, 2009.

[1230] Ivo Swartjes. *Whose Story Is It Anyway: How Improv Informs Agency and Authorship of Emergent Narrative*. PhD thesis, University of Twente, 2010.

[1231] Ivo Swartjes, Edze Kruizinga, and Mariët Theune. Let's pretend I had a sword: late commitment in emergent narrative. In *Proc. Joint International Conference on Interactive Digital Storytelling*, pages 264–267. Springer, 2008.

[1232] Ivo Swartjes, Edze Kruizinga, Mariët Theune, and Dirk Heylen. Emergent narrative and late commitment. In *Proc. Intelligent Virtual Agents*, pages 543–544. Springer, 2008.

[1233] Ivo Swartjes and Mariët Theune. A fabula model for emergent narrative. In *Proc. International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pages 49–60. Springer, 2006.

[1234] Ivo Swartjes and Joost Vromen. Emergent story generation: Lessons from improvisational theater. In *Proc. AAAI Fall Symposium on Intelligent Narrative Technologies*, pages 146–149, 2007.

[1235] Nicolas Szilas, Jean E Dumas, Urs Richle, Nicolas Habonneau, and Thomas Boggini. Nothing for dinner. `https://tbisim.unige.ch/portal`, 2014.

[1236] Nicolas Szilas and Ioana Ilea. Objective metrics for interactive narrative. In *Proc. International Conference on Interactive Digital Storytelling*, pages 91–102. Springer, 2014.

[1237] Nicolas Szilas, Olivier Marty, and Jean-Hugues Réty. Authoring highly generative interactive drama. In *International Conference on Virtual Storytelling*, pages 37–46. Springer, 2003.

[1238] Carole Tallant. Telling with, not telling to: Interactive storytelling and at-risk children. 1992.

[1239] K Tanaka, Hideyuki Nakashima, Itsuki Noda, Kôiti Hasida, Ian Frank, and Hitoshi Matsubara. MIKE: An automatic commentary system for soccer. In *Proc. International Conference on Multi Agent Systems*, pages 285–292. IEEE, 1998.

[1240] Joshua Glen Tanenbaum and Jim Bizzocchi. Close reading Oblivion: Character believability and intelligent personalization in games. *Loading...*, 3(4), 2009.

[1241] Grant Tavinor. What's my motivation? video games and interpretative performance. *The Journal of Aesthetics and Art Criticism*, 75(1):23–33, 2017.

[1242] Paul Taylor and Amy Isard. SSML: A speech synthesis markup language. *Speech Communication*, 21(1-2):123–133, 1997.

[1243] Team Bondi. La noire. *Rockstar Games*, 2011.

[1244] Hans ten Brinke, Jeroen Linssen, and Mariët Theune. Hide and sneak: Story generation with characters that perceive and assume. In *Proc. Artificial Intelligence in Interactive Digital Entertainment*, 2014.

[1245] Jonathan Teutenberg and Julie Porteous. Incorporating global and local knowledge in intentional narrative planning. In *Proc. International Conference on Autonomous Agents and Multiagent Systems*, pages 1539–1546. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[1246] Mariët Theune, Sander Faas, Anton Nijholt, and Dirk Heylen. The virtual storyteller: Story creation by intelligent agents. In *Proc. Technologies for Interactive Digital Storytelling and Entertainment*, volume 204215, 2003.

[1247] Frank Thomas, Ollie Johnston, and Walton Rawls. *Disney animation: The illusion of life*. Abbeville Press, 1981.

[1248] Wolfgang Thomas. Facets of synthesis: Revisiting Church's problem. In *Proc. International Conference on Foundations of Software Science and Computational Structures*, pages 1–14. Springer, 2009.

[1249] Sarah Grey Thomason and Terrence Kaufman. *Language contact, creolization, and genetic linguistics*. Univ. of California Press, 1992.

[1250] Thorny Games. *Dialect*. Thorny Games, 2017.

[1251] David Thue, Vadim Bulitko, and Marcia Spetch. Passage: A demonstration of player modeling in interactive storytelling. 8:227–228, 2008.

[1252] David Thue, Vadim Bulitko, Marcia Spetch, and Eric Wasylishen. Interactive storytelling: A player modelling approach. In *Proc. Artificial Intelligence in Interactive Digital Narrative*, pages 43–48, 2007.

[1253] David J Thue. *Generalized Experience Management*. PhD thesis, University of Alberta, 2015.

[1254] EA Tiburon. *Madden NFL 99*. EA Sports, 1998.

[1255] Sean Tiffee. The rhetorical alternative in neurocinematics. *M/C*, 20(1), 2017.

[1256] Pia Tikka. Neurocinematics: A filmmakerâĂŹs reflections of collaboration with neuroscientists. In *Intersecting Art and Technology in Practice*, pages 142–155. Routledge, 2016.

[1257] Pia Tikka. Systemic second order authorship for creating complex narratives—a neurophenomenological approach, 2017. Presented at the International Conference on Interactive Digital Storytelling.

[1258] Tzvetan Todorov. The categories of literary narrative. *Papers on Language and Literature*, 16(1):3–36, 1980 [1966].

[1259] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. Search-based procedural content generation: A taxonomy and survey. *Transactions on Computational Intelligence and AI in Games*, 2011.

[1260] John Ronald Reuel Tolkien. *On fairy-stories*. Oxford University Press, 1947.

[1261] Michael Tomasello and Hannes Rakoczy. What makes human cognition unique? from individual to shared to collective intentionality. *Mind & Language*, 18(2):121–147, 2003.

[1262] Boris Viktorovič Tomaševskij. Teoria literaturii: poetica. Gos. Izd., 1965 [1925].

[1263] John Marshall Townsend and Gary D Levy. Effects of potential partners' physical attractiveness and socioeconomic status on sexuality and partner selection. *Archives of Sexual Behavior*, 19(2):149–164, 1990.

[1264] Mike Treanor. *Investigating procedural expression and interpretation in videogames*. PhD thesis, University of California, Santa Cruz, 2013.

[1265] Mike Treanor and Michael Mateas. BurgerTime: A proceduralist investigation. In *Proc. Digital Games Research Association*, 2011.

[1266] Mike Treanor and Michael Mateas. Understanding representation in playable simulations. In *Proc. International Conference on the Foundations of Digital Games*, 2014.

[1267] Mike Treanor, Josh McCoy, and Anne Sullivan. Social play in non-player character dialog. In *Proc. Intelligent Narrative Technologies and Social Believability in Games*, 2015.

[1268] Mike Treanor, Josh McCoy, and Anne Sullivan. A framework for playable social dialogue. In *Proc. Artificial Intelligence and Interactive Digital Entertainment*, 2016.

[1269] Mike Treanor, Alexander Zook, Mirjam P Eladhari, Julian Togelius, Gillian Smith, Michael Cook, Tommy Thompson, Brian Magerko, John Levine, and Adam Smith. AI-based game design patterns. In *Proc. International Conference on the Foundations of Digital Games*, 2015.

[1270] TripMedia. *Burn:Cycle*. Philips Interactive Media, 1994.

[1271] Ann M Trousdale. Interactive storytelling: Scaffolding children's early narratives. *Language Arts*, 67(2):164–173, 1990.

[1272] John Truesdale, Sandy Louchart, Helen Hastie, and Ruth Aylett. Suitability of modelling context for use within emergent narrative. In *Proc. International Conference on Interactive Digital Storytelling*, pages 65–70. Springer, 2013.

[1273] Gaye Tuchman. Objectivity as strategic ritual: An examination of newsmen's notions of objectivity. *American Journal of Sociology*, 77(4):660–679, 1972.

[1274] Alan M Turing. Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65. Springer, 2009 [1950].

[1275] Scott R Turner. *MINSTREL, A story invention system*. PhD thesis, University of California, Los Angeles, 1985.

[1276] Scott R Turner. *Minstrel: a computer model of creativity and storytelling*. PhD thesis, University of California, Los Angeles, 1993.

[1277] Scott R Turner and Michael George Dyer. Thematic knowledge, episodic memory and analogy in MINSTREL, a story invention system. Technical report, 1986.

[1278] TV Tropes. Buddy cop show. `http://tvtropes.org/pmwiki/pmwiki.php/Main/BuddyCopShow`. Last accessed: June 18, 2018.

[1279] TV Tropes. Love triangle. `http://tvtropes.org/pmwiki/pmwiki.php/Main/LoveTriangle`. Last accessed: June 18, 2018.

[1280] TV Tropes. Reverse whodunnit. `http://tvtropes.org/pmwiki/pmwiki.php/Main/ReverseWhodunnit`. Last accessed: May 26, 2018.

[1281] Peter Tyson. *Getting Started with Dwarf Fortress: Learn to Play the Most Complex Video Game Ever Made*. O'Reilly Media, Inc., 2012.

[1282] Elke Uhl and Claus Zittel. Einleitung. weltprogrammierung—zur aktualität max benses. In *Max Bense. Weltprogrammierung*, pages 1–8. Springer, 2018.

[1283] Aarne Uotila. Procedural text generation with stateful context-free grammars. Master's thesis, University of Tampere, 2018.

[1284] Teun A Van Dijk. *Some aspects of text grammars: A study in theoretical linguistics and poetics*, volume 63. de Gruyter, 1972.

[1285] Bas C Van Fraassen. Semantic analysis of quantum logic. In *Contemporary Research in the Foundations and Philosophy of Quantum Theory*, pages 80–113. Springer, 1973.

[1286] Bas C Van Fraassen. Scientific representation: Paradoxes of perspective, 2010.

[1287] Taran Van Hemert and Trevor Van Hemert. Dwarf Fortress stories. `http://www.dfstories.com`. Last accessed: May 11, 2018.

[1288] Harriet van Horne. Funny computer—boon to TV. *The Pittsburgh Press*, Aug 1964.

[1289] Hugo van Kemenade. Love letters from a house of dust. In *Proc. History of Expressive Systems*. In press.

[1290] Guido VanRossum and Fred L Drake. *The Python language reference: Release 2.7.2*. Python Software Foundation Amsterdam, Netherlands, 2010.

[1291] Kevin Veale. *Comparing Stories-How Textual Structure Shapes Affective Experience in New Media*. PhD thesis, University of Auckland, 2012.

[1292] B Veld. Procedural generation of populations for storytelling. Master's thesis, Delft University of Technology, 2015.

[1293] Lois M Verbrugge. The structure of adult friendship choices. *Social Forces*, 56(2):576–597, 1977.

[1294] Paul Veyne. *Did the Greeks believe in their myths?: An essay on the constitutive imagination*. University of Chicago Press, 1988.

[1295] Bret Victor. `https://twitter.com/worrydream/status/620010357247979520`, Jul 2015.

[1296] Tal Vigderson. Hamlet ii: The sequel: The rights of authors vs. computer-generated read-alike works. *Loyola Law Review*, 28:401, 1994.

[1297] Shklovsky Viktor. Art as technique. In *Russian Formalism Criticism: Four Essays*, 1965.

[1298] Norvell E Von Behren. Word selecting device and method, 1968. US Patent 3,553,867.

[1299] Norvell E Von Behren. Word selecting device, 1971. US Patent 3,694,944.

[1300] Lev S Vygotsky. Mind in society: The development of higher psychological processes. 1978.

[1301] Marilyn A Walker, Ricky Grant, Jennifer Sawyer, Grace I Lin, Noah Wardrip-Fruin, and Michael Buell. Perceived or not perceived: Film character models for expressive NLG. In *Proc. International Conference on Interactive Digital Storytelling*, pages 109–121. Springer, 2011.

[1302] Marilyn A Walker, Owen Rambow, and Monica Rogati. Spot: A trainable sentence planner. In *Proc. North American Chapter of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics, 2001.

[1303] Marilyn A Walker, Jennifer Sawyer, Carolynn Jimenez, Elena Rishes, Grace I Lin, Zhichao Hu, Jane Pinckard, and Noah Wardrip-Fruin. Using expressive language generation to increase authorial leverage. In *Proc. Intelligent Narrative Technologies*, 2013.

[1304] Robin Walker, Gregory Coomer, and Erik Axel Johnson. Enhanced commentary system for 3d computer entertainment, 2014. US Patent 8,628,414.

[1305] J D Walsh. *Battleground (TV series*. Hulu, 2012.

[1306] Martin Walsh. Modeling perception and awareness in Tom ClancyâĂŹs Splinter Cell Blacklist. *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, page 313, 2015.

[1307] Richard Walsh. *The rhetoric of fictionality: Narrative theory and the idea of fiction*. The Ohio State University Press, 2007.

[1308] Richard Walsh. Emergent narrative in interactive media. *Narrative*, 19(1):72–85, 2011.

[1309] Derrick Ward and Henry Hexmoor. Towards deception in agents. In *Proc. International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1154–1155. ACM, 2003.

[1310] Noah Wardrip-Fruin. Three play effects–Eliza, Tale-Spin, and Sim City. *Digital Humanities*, pages 1–2, 2007.

[1311] Noah Wardrip-Fruin. *Expressive Processing: Digital fictions, computer games, and software studies*. MIT Press, 2009.

[1312] Noah Wardrip-Fruin. Digital media archaeology: Interpreting computational processes. In Erkki Huhtamo and Jussi Parikka, editors, *Media archaeology: Approaches, applications, and implications*. University of California Press, 2011.

[1313] Noah Wardrip-Fruin. Cmps 290j/danm 250d: Playable media. `https://courses.soe.ucsc.edu/courses/cmps290j/Winter14/01`, 2014. Last accessed: May 19, 2018.

[1314] Noah Wardrip-Fruin. Cmpm 290j/danm 250d: Playable media. `https://courses.soe.ucsc.edu/courses/cmpm290j/Winter15/01`, 2015. Last accessed: May 31, 2018.

[1315] Noah Wardrip-Fruin. We can and must understand computers now. In *Intertwingled*, pages 105–112. Springer, 2015.

[1316] Noah Wardrip-Fruin and Michael Mateas. Envisioning the future of computational media: the final report of the media systems project. Technical report, University of California, Santa Cruz, 2014.

[1317] Noah Wardrip-Fruin, Michael Mateas, Steven Dow, and Serdar Sali. Agency reconsidered. In *Proc. Digital Games Research Association*, 2009.

[1318] Stephen G Ware and R Michael Young. Intentionality and conflict in the best laid plans interactive narrative virtual environment. *Transactions on Computational Intelligence and AI in Games*, 8(4):402–411, 2016.

[1319] Allan Weallans, Sandy Louchart, and Ruth Aylett. Distributed drama management: beyond double appraisal in emergent narrative. In *Proc. International Conference on Interactive Digital Storytelling*, pages 132–143. Springer, 2012.

[1320] Eddy Webb. *Pugmire*. Pugsteady, In press.

[1321] Jonah Weiner. Where do dwarf-eating carp come from. *The New York Times Magazine*, Jul 2011.

[1322] Eugene A Weinstein. Toward a theory of interpersonal tactics. *Problems in social psychology*, pages 394–98, 1966.

[1323] Allen S Weiss. *Shattered forms: art brut, phantasms, modernism*. Student University of New York Press, 1992.

[1324] Paul Wells. ‘sorry blondie, i donâĂŹt do backstory!’ script editing: The invisible craft. In *Screenwriters and Screenwriting*, pages 151–169. Springer, 2014.

[1325] Rich Welsh. CrytekâĂŹs target tracks perception system. In Steve Rabin, editor, *Game AI Pro*, pages 403–411. CRC Press, 2013.

[1326] Peter William Weyhrauch. *Guiding interactive drama*. PhD thesis, Carnegie Mellon University, 1997.

[1327] Danny Weyns, Elke Steegmans, and Tom Holvoet. Towards active perception in situated multi-agent systems. *Applied Artificial Intelligence*, 18(9-10):867–883, 2004.

[1328] C. Albert White. *A history of the rectangular survey system*. US Department of the Interior, Bureau of Land Management, 1983.

[1329] Hayden White. The value of narrativity in the representation of reality. *Critical Inquiry*, 7(1):5–27, 1980.

[1330] Hayden White. The narrativization of real events. *Critical Inquiry*, 7(4):793–798, 1981.

[1331] Hayden White. *The content of the form: Narrative discourse and historical representation*. Johns Hopkins University Press, 2009.

[1332] Hayden V White. *Metahistory: The Historical Imagination in Nineteenth-century Europe*. Johns Hopkins University Press, 2014 [1973].

[1333] William J White, Jonne Arjoranta, Michael Hitchens, Jon Peterson, Evan Torner, and Jonathan Walton. Tabletop role-playing games. *Role-Playing Game Studies: Transmedia Foundations*, 2018.

[1334] Dan Whitehead. Systems vs. stories. *Eurogamer*, Jun 2013.

[1335] Nigel Whiteley. Banham and ‘otherness’: Reyner banham (1922–1988) and his quest for an architecture autre. *Architectural History*, 33:188–221, 1990.

[1336] Mads Wibroe, KK Nygaard, and P Bøgh Andersen. Games and stories. In Erik Granum, Berit Holmqvist, Søren Kolstrup, and K Halskov Madsen, editors, *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds*, pages 166–181. Springer Science & Business Media, 2013.

[1337] Jerry S Wiggins. *The five-factor model of personality: Theoretical perspectives*. Guilford Press, 1996.

[1338] Jerry S Wiggins. An informal history of the interpersonal circumplex tradition. *Journal of personality assessment*, 66(2):217–233, 1996.

[1339] Christopher Wilcock. An automated story generator of the Old French epic. Master's thesis, University of Sheffield, 2005.

[1340] Robert Wilensky. Points: A theory of the structure of stories in memory. *Strategies for Natural Language Processing*, pages 345–374, 1982.

[1341] Robert Wilensky. Story grammars versus story points. *Behavioral and Brain Sciences*, 6(4):579–591, 1983.

[1342] Robert M Wilson, Linda B Gambrell, and Warren R Pfeiffer. The effects of retelling upon reading comprehension and recall of text information. *The Journal of Educational Research*, 78(4):216–220, 1985.

[1343] Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

[1344] Gabriel Winslow-Yost. Simcity's evil twin. *The New Yorker*, Apr 2013.

[1345] Ludwig Wittgenstein. *Philosophical investigations*. John Wiley & Sons, 2009.

[1346] Rob Wittig and Mark C Marino. Occupy the emotional stock exchange, resisting the quantifying of affection in social media. *Humanities*, 6(2):33, 2017.

[1347] Daniel Wojcik. *Outsider Art: Visionary Worlds and Trauma*. University Press of Mississippi, 2016.

[1348] Abraham Wolf. The philosophy of probability. In *Proc. Aristotelian Society*, volume 13, pages 328–361. JSTOR, 1912.

[1349] Thomas H Wolf. Letter to J. Francis Reintjes, Mar 1960. Douglas T. Ross papers. MC0414, box 217. Institute Archives and Special Collections, MIT Libraries, Cambridge, Massachusetts.

[1350] Tom Wolfe and E W Johnson. The new journalism. 1973.

[1351] Pree Wolfgang. Design patterns for object-oriented software development. *Reading Mass*, page 15, 1994.

[1352] John Woods. Considérations sémantiques sur la logique de la fiction. *Dialogue: Canadian Philosophical Review/Revue canadienne de philosophie*, 12(1):50–61, 1973.

[1353] John Woods. Fictions and their logic. In *Philosophy of Logic*, pages 1061–1126. 2007.

[1354] John Woods. *Truth in Fiction: Rethinking Its Logic*, volume 391. Springer, 2018.

[1355] John Hayden Woods. The logic of fiction: A philosophical sounding of deviant logic. 1974.

[1356] William A Woods, Ronald M Kaplan, and Bonnie Nash-Webber. *The lunar sciences natural language information system*. Bolt, Beranek and Newman, 1972.

[1357] R M Worthy. Librascope Division of General Precision, Inc. In *Current Research and Development in Scientific Documentation*, volume 9, pages 195–196. Nov 1961.

[1358] R. M. Worthy. A new American poet speaks: The works of A.B. *Horizon: A Magazine of the Arts*, IV(5):96–99, 1962.

[1359] Steven T. Wright. How the mixed reality game 'Bad News' brings towns like 'Twin Peaks' to life. *Rolling Stone*, Apr 2017.

[1360] Andrew J Wu. From video games to artificial intelligence: Assigning copyright ownership to works generated by increasingly sophisticated computer programs. *AIPLA Quarterly Journal*, 25:131, 1997.

[1361] Georgios N Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. Player modeling. In *Dagstuhl Follow-Ups*, volume 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[1362] Adrienne Yapo and Joseph Weiss. Ethical implications of bias in machine learning. In *Proc. Hawaii International Conference on System Sciences*, 2018.

[1363] Masoud Yazdani. How to write a story. In *Proc. 5th European Conference on Artificial Intelligence*, pages 259–260. North-Holland, 1982.

[1364] Masoud Yazdani. Generating events in a fictional world of stories. Technical Report R. 113, Department of Computer Science, University of Exeter, 1983.

[1365] Masoud Yazdani. Reviewing as a component of the text generation process. In *Natural Language Generation*, pages 183–190. Springer, 1987.

[1366] Masoud Yazdani. Computational story writing. pages 125–147. 1989.

[1367] Masoud Yazdani. ROALD—a framework for story writing. `https://www.intellectbooks.co.uk/masoud/author/story/story-3.htm`, undated.

[1368] Victor H Yngve. Random generation of english sentences. In *Proc. Machine Translation of Languages and Applied Language Analysis*, 1961.

[1369] R Michael Young. Computer support for collaborative dramatic art. In *Working notes of the Workshop on Use and Design of MUDs for Serious Purposes*, 1996.

[1370] R Michael Young. Notes on the use of plan structures in the creation of interactive plot. In *Proc. AAAI Fall Symposium on Narrative Intelligence*, pages 164–167, 1999.

[1371] R Michael Young. Science considered helpful. In *Proc. International Conference on Interactive Digital Storytelling*, pages 21–35. Springer, 2018.

[1372] R Michael Young, Stephen G Ware, Brad A Cassell, and Justus Robertson. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, 37(1-2):41–64, 2013.

[1373] Raymond De Young and Martha C Monroe. Some fundamentals of engaging stories. *Environmental Education Research*, 2(2):171–187, 1996.

[1374] Serena Zampolli. Grammar Stories: A proposal for the narrativization of abstract contents. In *Proc. International Conference on Interactive Digital Storytelling*, pages 388–391. Springer, 2017.

[1375] Serena Zampolli. Verb+s is looking for love: Towards a meaningful narrativization of abstract content. In *Proc. International Conference on Interactive Digital Storytelling*, pages 126–134. Springer, 2017.

[1376] Maliang Zheng and Daniel Kudenko. Automated event recognition for football commentary generation. *International Journal of Gaming and Computer-Mediated Simulations*, 2(4):67–84, 2010.

[1377] Jichen Zhu and D Fox Harrell. Daydreaming with intention: Scalable blending-based imagining and agency in generative interactive narrative. In *Proc. Creative Intelligent Systems*, 2008.

[1378] Richard Ziegfeld. Interactive fiction: A new literary genre? *New Literary History*, 20(2):341–372, 1989.

[1379] Eric Zimmerman. Against hypertext. *American Letters & Commentary*, 12, 2000.

[1380] Zoesis Studios. *The Penguin Who Wouldn't Swim*. Zoesis, Inc, 1999.

[1381] Alexander Zook. *Automated iterative game design*. PhD thesis, Georgia Institute of Technology, 2016.

[1382] Robert Zubek. *Hierarchical parallel Markov models for interactive social agents*. PhD thesis, Northwestern University, 2005.

[1383] Robert Zubek. Needs-based AI. *Game Programming Gems 8*, page 302, 2010.

[1384] Robert Zubek. 1000 NPCs at 60 FPS. In Steve Rabin, editor, *Game AI Pro 3: Collected Wisdom of Game AI Professionals*. CRC Press, 2017.

[1385] Lisa Zunshine. *Why we read fiction: Theory of mind and the novel*. Ohio State University Press, 2006.