Title
Programming Insight: Human and Machine Intelligence in the Petabyte Age

Author
Udekwu, Osita Anders

Publication Date
2017

Peer reviewed|Thesis/dissertation

Programming Insight: Human and Machine Intelligence in the Petabyte Age

by

Osita Anders Udekwu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Rhetoric

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor David Bates, Chair
Professor Charis Thompson
Professor Michael Wintroub

Summer 2017

Abstract

Programming Insight: Human and Machine Intelligence in the Petabyte Age

by

Osita Anders Udekwu

Doctor of Philosophy in Rhetoric

University of California, Berkeley

Professor David Bates, Chair

This dissertation, entitled *Programming Insight: Human and Machine Intelligence in the Petabyte Age*, explores the ideal of "intelligent" organizations through a close reading of a data fusion and analysis environment for government and enterprise by Palantir Technologies. Using approaches from science and technology studies, human-computer interaction, and new media theory, the project links understandings of computation and interactivity with emerging infrastructures for knowledge practices. My work examines the relationships between epistemology, ethics, and aesthetics for a digital platform that works in support of organizations that, in the words of former NSA director Michael Hayden, "kill people based on metadata."

My dissertation examines the digital and human processes that purport to transform digital traces into knowledge and insight in domains from finance to counter-terror. Commonly cast as a generic surveillance technology, I argue that Palantir's products index a range of technical and cognitive performances that allow analysts and their organizations to see digital traces as constitutive of agential activities like data exfiltration, fraud, and terrorism. Through an analysis of publicly available materials including demonstration videos, conference proceedings, technical papers, and patent filings, I present the rhythms of a digital environment that elicits a hybrid investigative intelligence in contexts where data are presumptively and overwhelmingly computerized. Drawing on contemporary and historical work in science and technology studies and human-computer interaction, the project traces these instances of knowledge production as composite epistemic performances among humans and machines. The environment and interface for these performances are constructed as a procedurally designed site for improvised interactions – as a space of play. What results are investigative narratives that name, explain, and target instances of elusive, anomalous, or threatening behavior mediated by digital systems. These structuring categories, however, are neither fundamental nor disinterested, but emerge from highly cathected figures of organizational risk including whistleblowers, malicious insiders, cybercriminals, and terrorists. I contend that such data-based, organizational "intelligence" is a constructed effect of distributed cognitive assemblages, and that a variety of agents and epistemic mediators – from machine learning algorithms, to graph visualizations, to experienced human analysts – shape the resulting possibility space, the concepts invoked, and the knowledge produced.

The project contributes to critical work on new media, risk, and expertise, as I analyze how recent Big Data technologies present these risks as networked phenomena rendered from massive sets of relational digital data. In particular, I argue that contemporary critiques of risk management must engage with a growing domain of problems and techniques centered on the detection of anomalies. The challenges of anomaly detection encompass not only the identification and management of risk to prevent negative outcomes, but also related opportunities for the exploitation of anomalies, most prominently the pricing of assets in financial markets. The mechanics of knowledge discovery and data mining (KDD) systems and how they represent and recirculate anomaly and risk – among organizations that alternately seek to exploit the risk profiles of others and minimize their own – brings studies of digital media interfaces and infrastructure into conversation with science and technology studies and political thought. The construction of such systems may signal a shift away from the nation-state as the center of technopolitical critique, as a national government or agency becomes one customer and/or vendor in a larger field of non-democratic firms and organizations exchanging access to datasets, analysis tools, and other technologies of modern governance.

Contents

"The world's hardest problems": Palantir and the ontology of intelligence

As of 2017, the Palo Alto, California company Palantir Technologies, Inc. has emerged as an exemplar of the contemporary surveillance tools that have grown out of developments in data fusion, analysis, and machine learning techniques. A private company founded in 2004 by PayPal co-founder and venture capitalist Peter Thiel, Palantir also counts In-Q-Tel, the venture capital firm associated with the United States intelligence community, among its early investors, providing $2 million for the company's initial round of funding.[1] Current estimates put Palantir's value at $20 billion.[2] In addition to the United States CIA, Palantir has counted among its clients the New York and Los Angeles Police Departments, JP Morgan Chase, and SAC Capital Advisors.[3] More recently, Palantir was found to have obtained a $41 million contract (award in 2014) to develop a new intelligence system for United States Immigration and Customs Enforcement, with a platform bringing together data from a variety of government databases to not just manage cases, but discover new targets.[4] A June 2014 *New York Times* article claimed Palantir's software "has illuminated terror networks and figured out safe driving routes through war-torn Baghdad," and "has also tracked car thieves, helped in disaster recovery, and traced salmonella outbreaks."[5] In 2011, the company was the subject of a brief scandal surrounding a leaked presentation given to Bank of America on a collaboration to target the WikiLeaks organization, for which CEO Alex Karp publicly apologized.[6] Karp notes that many of Palantir's contracts have "massive NDA [non-disclosure agreement] clauses,"[7] but the company has embraced, to some degree, the open culture touted at many Silicon Valley startups, even organizing conferences demonstrating its products and discussing the challenges facing partner organizations in finance, information security, consumer banking, law enforcement, insurance, and pharmaceutical research. Talks at these conferences included not just technical talks, but more philosophical lectures with titles like "Humans, Data, and the Culture of Too Much Information" and "Friction in the Machine: How Fluid Processes Allow Optimal Human-Computer Interaction."[8]

Designed for organizational clients, corporations and government agencies, Palantir's products help "clients manage and analyze massive amounts of information from a host of

---

[1] Heather Somerville, "Uber valued at whopping 18.2 billion," *San Jose Mercury News*, June 6, 2014, http://www.mercurynews.com/business/ci_25913159/18-2-billion-valuation-makes-uber-top-venture.

[2] Oliva Zaleski, "WeWork Becomes World's Fifth Most-Valuable Startup," *Bloomberg Technology*, July 12, 2017, https://www.bloomberg.com/news/articles/2017-07-12/wework-is-said-to-become-world-s-fifth-most-valuable-startup.

[3] Gregory Maus, "A (Pretty) Complete History of Palantir," August 11, 2015, http://www.socialcalculations.com/2015/08/a-pretty-complete-history-of-palantir.html

[4] Spencer Woodman, "Palantir Provides the Engine for Donald Trump's Deportation Machine," *The Intercept*, March 2, 2017, https://theintercept.com/2017/03/02/palantir-provides-the-engine-for-donald-trumps-deportation-machine/.

[5] Quentin Hardy, "Unlocking Secrets, if Not Its Own Value," *New York Times*, June 1, 2014, http://www.nytimes.com/2014/06/01/business/unlocking-secrets-if-not-its-own-value.html.

[6] Maus, "A (Pretty) Complete History of Palantir."

[7] Alex Karp, "Interview: Alex Karp, Founder and CEO of Palantir," interview by Evelyn Rusli, YouTube video, 6:40, posted by "TechCrunch," February 16, 2012, https://www.youtube.com/watch?v=VJFk8oGTEs4.

[8] Palantir Technologies, "Human, Data, and the Culture of Too Much Information," YouTube video, 23:37, posted by "Palantir," June 4, 2013, https://www.youtube.com/watch?v=P2tsS4ktt30; Palantir Technologies, "Friction in the Machine: How Fluid Processes Allow Optimal Human-Computer Interaction," YouTube video, 38:37, posted by "Palantir," November 4, 2011, https://www.youtube.com/watch?v=Kw1hZkfOVhQ.

sources."[9] As for self-description, in June 2014 Palantir's website claimed that they "help solve the world's hardest problems," and described Palantir as a company that "make[s] products for human-driven analysis of real world data."[10] Palantir does not produce software for data visualization or data mining.[11] Rather, they are building a software platform, an analysis infrastructure "that enables people to take whatever data is relevant to them and understand it more easily and thoroughly than ever before, using concepts that they already understand."[12] It is useful to note that Palantir is not in the business of data collection or data analysis, though they have advertised certain products in partnership with data providers (e.g., QA Studio, a version of their Metropolis/Finance platform paired with a subscription to market data from Thomson Reuters). Instead, Palantir engineers "deploy" instances of their platform on hardware sited at client organizations and assist those clients with the integration and adaptation of their existing data assets into the Palantir deployment.

The ambition for the Palantir platform is "conceptual analysis at the speed of thought."[13] The company's name, Palantir, refers to a set of mystical orbs from J.R.R. Tolkien's *The Lord of the Rings*, artifacts that allow a capable user to see events distant in space and time.[14] References to the Tolkien universe (including the informal name for their Palo Alto headquarters, "The Shire"), abound in the company's culture, and this evocation of a boundless, oracular vision is striking. This promise of vision is connected with their characterization of analysis as "everything necessary to extract **insight** from **information** [emphasis in source]."[15] This analytical power is something they persistently locate with the human analyst or expert rather than machines, and Palantir describes their mission as providing "technology that preserves the essential role of human judgment and individual responsibility in big data analytics, and to do so through design and engineering practices that augment sound decision making rather than artificially shifting or displacing what already works."[16] Their platform attempts to build human domain expertise back into data analysis systems because, in their view, although algorithms can provide basic inferences, "there's no algorithm that replaces human intuition."[17]

Centering on an analysis of publicly available materials describing the interactive data analysis platforms from Palantir Technologies, this dissertation examines the technical processes that purport to transform data into knowledge, then knowledge into insight. With the expanding generation and collection of digital traces naturalized as the foundation of both the private and

---

[9] Peter Delevett, "Palantir Technologies' latest funding round could top $200 million," *San Jose Mercury*, September 27, 2013, http://www.mercurynews.com/2013/09/27/exclusive-palantir-technologies-latest-funding-round-could-top-200-million/.
[10] "About | Palantir," Palantir Technologies, accessed June 27, 2014, https://www.palantir.com/about/. Page modified, archived version from February 15, 2014, https://web.archive.org/web/20140215172616/http://www.palantir.com/about/.
[11] Kevin Simler, "Palantir: so what is it you guys do?" *The Palantir Blog*, December 4, 2007, accessed May 9, 2016, https://palantir.com/2007/12/what-do-we-do. As of writing, this page has been removed. A link to archived versions of this page (and other removed pages) can be found in the "Palantir sources" section preceding the bibliography.
[12] Ibid.
[13] Ari Gesher, "Palantir: embodying a 50-year-old vision of the future?" *The Palantir Blog*, March 16, 2007, accessed May 9, 2016, https://palantir.com/2007/03/human-computer-symbiosis. (Page removed.)
[14] Robert Foster, *The Complete Guide to Middle-Earth: From the Hobbit through The Lord of the Rings and beyond* (New York: Ballantine Books, 2001), 396.
[15] Simler, "So what is it you guys do?"
[16] John Grant, "Datafication and You," *The Palantir Blog*, October 25, 2013, accessed May 9, 2016, https://palantir.com/2013/10/datafication-and-you. (Page removed.)
[17] Palantir Technologies, "Human, Data, and the Culture of Too Much Information," YouTube video.

public organizational behavior, I examine the implicit portraits of organizations inhabiting a digital, data-based reality. Palantir's use in domains such epidemiology and public health, drug discovery, finance, intelligence, and disaster response points to an expansive conceptualization of security and risk that encompasses a broad understanding of abnormality, disruption, involving human and non-human actors alike. Attention to the role of algorithms in governance and classification, I argue, should respond to a growing number of systems for human-driven investigation where machine classification may not play the dominant role. In this introduction, I begin by exploring the intellectual and technical frameworks that support Palantir's claims of applicability across sectors, ultimately a claim of their data-based commensurability. In other words, what are "the world's hardest problems"? What do they have in common? On what terms does Palantir claim to succeed where others have failed?

Anomaly detection: How not to get fleeced by the Russian mob

Palantir locates the origins of their unique approach to data analysis in the fraud detection unit at the online payment portal PayPal, who, as with other early competitors, was faced with the challenge of discerning genuine from fraudulent transactions on their platform, to do business "without getting fleeced by the Russian mob."[18] According to Palantir, what allowed PayPal to survive in this space was to turn away from the algorithmic or automated approaches its competitors took to this analysis problem. Rather, the PayPal fraud team focused on empowering and increasing the effectiveness of their human analysts. They saw this design approach as applicable to many domains, and in 2004 Peter Thiel funded the creation of a prototype for Palantir by a PayPal engineer and two Stanford graduate students. This initial use case, fraud detection, provides a point of departure for understanding Palantir's diverse problem spaces and to begin specifying their claim to address "the world's hardest problems." In computer science, payment fraud is a classic anomaly detection problem, a category that includes issues in network and systems monitoring (intrusion), insurance claims (fraud), and industrial systems (damage or sabotage).[19] As a field of applied research, anomaly detection draws on computer science, mathematics, statistics, and engineering to articulate and address a broad set of what could be characterized as security problems, including unauthorized use of computer systems, fraudulent financial activity, and the activity of criminal organizations. Anomaly detection techniques are also related to efforts to anticipate profitable trading opportunities (algorithmic and high-frequency trading), as well as the emergence of macro-scale events like market crashes and disease outbreaks.[20] The prominent use cases put forward in Palantir's archive (its patents, demonstration and conference videos, blog posts, and web site) include many of these problem spaces, presenting a link to this broader class of systems. It seems that "the world's hardest problems" are problems of security and, in particular, of anomaly detection.

Anomaly detection systems are an increasingly important class of political technologies. The deployment of these systems cuts across earlier distinctions between social, economic, and

---

[18] Palantir Technologies, "GovCon7: Introduction to Palantir," YouTube video, 40:01, posted by "Palantir," November 2, 2011, https://www.youtube.com/watch?v=f86VKjFSMJE.

[19] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)* 41, no. 3 (2009): 15.

[20] Chandola et al., "Anomaly detection"; Steve Lohr, "A Data Weapon to Avoid the Next Financial Crisis," *The New York Times Bits Blog*, September 9, 2013, https://bits.blogs.nytimes.com/2013/09/09/a-data-weapon-to-avoid-the-next-financial-crisis/.

political domains, and a new kind of commensurability emerges among knowledge practices that surround citizens, markets, employees, enemy combatants, and criminals. The concept of the signature, for instance, has been highly productive for the development of network intrusion detection and other cybersecurity systems. The calculation of short but distinctive concatenations or traces from larger chunks of data allows for a quick comparison of files, process behavior, and network activity with catalogs of previously classified examples – either legitimate or malicious. An early argument for the productive link between the concept of signatures and the identification of anomalies was put forward in the work of Forrest, et al. on conceptions of "self and other" for the securing UNIX systems.[21] Forrest's further work drew inspiration from the distinct "tags" produced by T-cells in immune systems and resulted in a consequent push for the embrace of immunology as a guiding analogy for research in computer security.[22] This fixation on the power of recognizing small but unique traces of content, for instance in telephone metadata, also informs the logic of the United States government's "signature strikes" by drones in its War on Terror. With the increasing digital mediation of social life, there is a growing convergence both in the ways organizations collect data about behavior across domains and in their methods for distinguishing between what is normal and what is anomalous, illicit, or threatening.

Organizations come to know the world through digital data, but this data must be meaningful by additional systems, and these systems make that meaning or attempt to do so through their technical articulation of certain metaphorical understandings of their target domains. An assertion of anomaly takes place with respect to a background of a presumed normal behaviors, and techniques for anomaly detection are often categorized based on the associated method for characterizing the domain along with expected or normal data points. The survey of anomaly detection by Chandola et al. groups techniques based on the assumptions each makes about the nature of normal and anomalous data points. Nearest neighbor-based techniques use distance measures and assume that "normal data instances occur in dense neighborhoods, while anomalies occur far from their closest neighbors."[23] Similarly, clustering-based techniques assume that normal data instance belongs to a cluster, while anomalies do not.[24] Information theoretic techniques assume that anomalies "induce irregularities in the information content of the data set," for instance, a subset of data that significantly raises the complexity of the overall data set.[25] Each set of techniques depends on the translation and naturalization of these metaphors, the making of metaphors into algorithms, algorithms which make sense of the datasets resulting from the correlated translation of the world into digital traces. Moreover, each set has its own deficiencies when it comes to the need for training data sets, human supervision of training, computational costs, and adequacy to a given use case. Put otherwise, the effectiveness of a data analysis system results from the choice of metaphor, the efficiency of the metaphor's technical articulation, and the relevance its outputs. In this context, Palantir's gambit is to propose a different relationship between these elements, one that shifts the decisive

---

[21] Stephanie Forrest, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of 1994 IEEE Symposium on Research in Security and Privacy* (IEEE, 1994): 202-212.
[22] Stephanie Forrest, Steven A. Hofmeyr, and Anil Somayaji, "Computer immunology," *Communications of the ACM* 40, no. 10 (1997): 88-96.
[23] Chandola et al., "Anomaly detection," 15:22.
[24] Ibid., 15:27.
[25] Ibid., 15:35-36.

moments of epistemic performance away from the computer and toward the human expert: "We believe in augmenting human intelligence, not replacing it."[26]

The development of systems for the detection and investigation of anomalies, using automatic or symbiotic techniques, reveals a field of practice for organizations in both government and enterprise centered on elusive risks whose traces are hidden in streams of normal data. A preoccupation with notions like anomaly – case, risk, danger, crisis – is a feature of what Michel Foucault called the apparatuses of security that emerged around the new "political technology" of *population*.[27] Foucault's analysis of these terms as objects of "critical reflection" for ministers, scholars, and other thinkers faced with new and specific problems – the market town, grain scarcity, epidemics – presents a method for understanding the role of data analysis systems and paradigms in the contemporary moment. Attention to the perceived deficiencies and obstacles within those reflections on practice reveals some of the heterogeneous assemblages and rationalities that constitute governmentality. Moreover, the location of some of these reflections outside of their previous state context attests to the potential overvaluation of the state in the analysis of power and control. While the neoliberal withering of the state, beginning in the late 20th century, is an oft-discussed topic, we should be alert to the potential of devolution of state power and its relocation to other organizational, especially corporate, sites. What John Cheney-Lippold calls the "soft biopolitics" of web tracking and analytics companies, for instance, is compatible but not identical with state-sponsored surveillance programs.[28] Indeed, computer security expert Bruce Schneier argues that governments merely "piggyback" on the capabilities developed by enterprise, but corporate entities, however, are not always quick to comply.[29] Emerging from this is a post-Foucauldian portrait of power that requires a topological analysis, foregoing any epochal readings of the logic of government or, for that matter, the privileging of the state within that logic.[30]

Anomaly detection, then, comprises set of technological reflections on the problems of the organization as problems of digitally mediated knowledge of risk, threats, and opportunities, casting computing machinery as a source of necessary "intelligence" for a diverse array of organizations. Surveying these techniques and use cases reveals an important set of objectives and horizons for reflective organizational practice across government and enterprise, entangling the design and deployment of computerized data analysis systems with our conceptualizations of what it means to be a modern organization, whether that organization is focused on counterterror, public health, or consumer banking. What seem to be technical debates about efficacy also indicate political assumptions about the nature, urgency, and appropriateness of the objects and tools of knowledge and intervention for these organizations. Based on the sketch above, one might examine, for instance, of the intrication of governmentality with techniques for the generation of "signatures" for user behavior on computer networks and other sociotechnical systems in the pursuit of anomalies. This would reveal a host of heterogeneous sites and objectives, reflecting new contours, legitimacies, visibilities, and points of contestation across

---

[26] "About | Palantir," Palantir Technologies, accessed July 17, 2017, https://palantir.com/about/index.html.

[27] Michel Foucault, *Security, Territory, Population: Lectures at the Collège de France, 1977-78*, ed. Michel Senellart, trans. Graham Burchell (New York: Palgrave Macmillian, 2007): 61.

[28] John Cheney-Lippold, "A New Algorithmic Identity: Soft Biopolitics and the Modulation of Control," *Theory, Culture & Society* 28, no. 6 (2011): 164-181.

[29] Bruce Schneier, "NSA Surveillance: What We Know, and What to Do about It," YouTube video, 53:39, posted by "RSA Conference," May 5, 2014, https://www.youtube.com/watch?v=1iMFPMqboZc.

[30] Stephen Collier, "Topologies of Power: Foucault's Analysis of Political Government beyond 'Governmentality'," *Theory, Culture & Society* 26, no. 6 (2009): 78-108.

social, economic, and political life (insofar as we separate them). Palantir's design and its embrace of human computer symbiosis in response to anomaly detection problems represents another "turn" in critical reflection on "the world's hardest problems," and we can infer from its technical arrangements (both in the techniques are endorsed and those rejected) aspects of contemporary topologies power: the dispersed sites; the common genres of technical and organizational apparatus; and what other objects and horizons may remain unannounced or unrealized.

Approaching Analysis and Anomaly Detection

In their discussions of the design philosophy behind the Palantir platform, their engineers persistently (and explicitly) invoke J.C.R. Licklider's discussion of "man-computer symbiosis" and "friction" as a key point of departure.[31] Palantir's stated design goal is shifting the focus away from computation and algorithm to interaction and usability. Palantir provides an "analysis infrastructure" that reframes anomaly detection problems in terms of composite intelligent systems: human and computer together with a low-friction interface between, emphasizing the human as the source of the decisive cognitive contributions. The Palantir archive frequently illustrates this philosophy with a story about the 2005 PAL/CSS Freestyle chess tournament where an amateur player defeated a purpose-built chess computer and a computer-aided grandmaster. This amateur "player," named "ZackS" was actually two human players running four chess engines on three different laptops. Russian grandmaster Garry Kasparov glosses the insight from this example: "Weak human + machine + better process was superior to a strong computer and, more remarkably, superior to a strong human + machine + inferior process."[32] Palantir does not speak much to what actually constituted the "better process" for ZackS and the chess context, but for their purposes, the "most central hard problem […] in trying to enable the analyst is data modeling." Data models, or ontologies, are Palantir's first response to the question of "how to bring vast amounts of information into productive contact with human intelligence." With this, I think the example of ZackS suggests a common (if unacknowledged) theme in Palantir's interface and design – friction is a necessary aspect of the contact between heterogeneous elements in these composite arrangements, but this contact is ultimately productive. What should not be overlooked in ZackS is the number of elements, the proliferation of points of contact between elements of different kinds. With respect to the problem of data modeling, this recasts the key question as how to best enculture digital traces such that human contact with them produces knowledge.[33]

Both logically and visually, Palantir installs digital traces in "a new phenomenal field defined by social agents," the programmers and designs of Palantir and their industry partners. In Palantir environments, analysis becomes experimental and displays features characteristic of Karin Knorr Cetina's definition of the laboratory, full of instruments for the detachment and manipulation of objects in an "enhanced" environment, one where previously elusive

[31] J.C.R. Licklider, "Man-Computer Symbiosis," *IRE Transactions on Human Factors in Electronics* 1 (1960): 4-11.
[32] Garry Kasparov, "The Chess Master and the Computer," *The New York Review of Books*, February 11, 2010, http://www.nybooks.com/articles/2010/02/11/the-chess-master-and-the-computer/.
[33] Karin Knorr Cetina, *Epistemic Cultures: How the Sciences Make Knowledge* (Cambridge: Harvard University Press, 1999).

characteristics and relationships can become apparent.[34] The analytical capability of these systems emerges from their "ongoing work of instituting specific differences from which epistemic dividends can be derived," and we might wonder what differences are emphasized in the ZackS chess assemblage versus that of the purpose-built chess computer and chess grandmasters they defeated.[35] Karin Knorr Cetina's theorization of the laboratory and its reconfiguration of natural and social orders entails that epistemic processes are also ontological processes, and knowledge is the contingent product of alignments, assemblages, and networks of people and different kinds of things. The perspective of what Charis Thompson calls the "ontological branches" of science technology studies emphasizes the ways that certain moments of knowledge emerge from the maneuvering of objects and kinds, that epistemic apparatuses are highly interventional and do not simply reflect a world already in place.[36] Computing machinery does not present a passive surface for the acceleration of human intellect, but rather architectures for interaction that create new epistemic processes unto themselves. These architectures are, at first, glance semantic and visual. They must also be seen as the outcome of pre-categorical and/or proto-categorical modulations within the machine, specific differences that do not appear to us as such, only together as they are constantly summed and sequenced. Human-computer interaction in epistemic processes invokes ontology both as a technical and a philosophical concept and requires an account for the reality of the digital and its functions, abstractions, and processes. In particular, my account of Palantir will pay close attention to the ways in which data "objects" are produced, managed, and manipulated as part of analysis.

Karen Barad amplifies these insights from the ontological branches of STS and argues further that the deployment of the ontological processes in science and technology do not just "make" our facts, our natural and social categories, but are part of the making of objects and subjects of such knowledge in the first place. Bringing together the work of Judith Butler and physicist Niels Bohr's "relational ontology," Barad claims that our perceptions and discursive knowledge of delimited, material objects are a species of performative effect. Our conventional understandings of independent, bounded, propertied objects are actually the result of an "agential cut" that "enacts a local causal structure among 'components' of a phenomenon in the marking of the 'measuring agencies' ('effect') by the 'measured object' ('cause')."[37] Barad's work pushes not only for the epistemological inseparability of the knowing agent and the object known but also for their ontological inseparability.[38] Things like 'objects' and 'agents' are only temporary stabilizations of "ontologically primitive relations – relations without pre-existing relata," generic and undifferentiated "phenomena."[39] As Barad argues, agency is not "an attribute of 'subjects' or 'objects,'" rather, it is "a matter if *intra-activity* [emphasis mine]; it is something that happens in process, not something that someone or something has."[40] Barad's agential realism invites a discussion of how digital data collection and analysis, as epistemic performances, constitute persons and organizations as knowing, intelligent agents and, in concert, the objects and domains they claim to analyze and know. Intelligence, that seemingly

---

[34] Ibid., 26-27.
[35] Ibid., 44.
[36] Charis Thompson, *Making Parents: The Ontological Choreography of Reproductive Technologies* (Cambridge: The MIT Press, 2005), 47.
[37] Karen Barad, "Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter," *Signs* 28 (no. 3, 2003): 815.
[38] Ibid.
[39] Ibid.
[40] Ibid., 826-827.

elusive property of a unique class of entities, would be better understood as a peculiar rhythm of intra-activity, characterizing an ensemble of actors and apparatuses in processes of 'thingification' and 'subjectification.'

Speculating on the actual processes taking place among the multi-human, multi-computer chess team ZackS, a Palantir speaker off-handedly refers to the "crazy ballet" that must have taken place.[41] This comment is somewhat revelatory, suggestive as it is of a kind of choreography between people and things, or rather, as Charis Thompson terms it, an *ontological choreography* that entails "the grafting of parts and the calibrating of time."[42] Digital data analysis platforms display rhythms of representation and transformation with respect to data "objects," with the goal of leveraging the cognitive timings of human analysts with each other (via collaboration and information sharing tools) and with those of the computerized database. Though Palantir frequently deploys J.C.R. Licklider's concepts around "friction" which emphasize the quantitative acceleration of an existing analysis process, Thompson's concept of choreography provides a viewpoint that includes the qualitatively productive rhythms of contact and separation that occur amongst heterogeneous elements. Within computing machinery and across its interfaces, it is not only about the institution of specific differences but their sequence and timing that yield epistemic dividends.

Analyzing the Palantir platform: Reverse Engineering and Design Recovery

Karen Barad's work would suggest that in examinations of digital media the visual, discursive human-computer interface is only one relevant site for analysis, as the interface is a *material* encounter with a digital object – and this encounter will be structured not just by graphic elements on-screen, but also by the programmed opening and closing of semiconductor gates. Computer code is one way in which "the material and the discursive are mutually implicated in the dynamics of intra-activity," though its ethereality also tempts us to isolate, analytically, a discursive surface and forget the inscrutable or distant materiality of the devices.[43] The task of elaborating the structure of these encounters and performances, their technicity, and their networks of effects I would frame as an instance of *reverse engineering*. While I am, in the first instance, prompted by Wolfgang Ernst's (somewhat cryptic) description of his methodology of media archaeography as "epistemological reverse engineering,"[44] the technical discourses of reverse engineering and design recovery (a related term, one more common in discussing software systems) supply an outline for a practice of reading and writing about digital artifacts, one that can ultimately help link the technicity of these systems with their effects beyond the interface.

Michael Rekoff defines reverse engineering as "the act of creating a set of specifications for a piece of hardware by someone other than the designers, primarily based upon analyzing and dimensioning a specimen or collection of specimens."[45] Such efforts are often directed at the

---

[41] Palantir Technologies, "Human-Computer Symbiosis: Efficient and Powerful Use of Computing Power," YouTube video, 37:43, posted by "Palantir," November 3, 2010, https://www.youtube.com/watch?v=oLalkcMDCwg.
[42] Thompson, *Making Parents,* 9.
[43] Barad, "Posthumanist Performativity," 822.
[44] Wolfgang Ernst, "Media Archaeography," in *Media Archaeology: Approaches, Applications, and Implications,* ed. Erkki Huhtamo and Jussi Parikka (Berkeley: University of California Press, 2011), 239.
[45] Michael G. Rekoff, "On reverse engineering," *IEEE Transactions on systems, man, and cybernetics* 2

creation of clones or surrogates for the hardware in question, or to "overcome defects in or to extend the capabilities of existing apparatus."[46] As an organizational practice, these techniques are frequently used in the maintenance and replacement of "legacy systems" – especially software systems – where original design information, whether from humans or documentation, is insufficient or has been lost altogether. Of course, reverse engineering is not limited to maintenance of systems at legitimate business or governmental organizations. Reverse engineering is a practice commonly associated with corporate and military espionage, as well as consumer hobbyists and do-it-yourself enthusiasts. The 'hacking' of consumer devices has increasingly become a target for legislation in the 21st century, as companies attempt to both protect intellectual property and profit from monopolies on the upgrade and maintenance of their products. At first glance, reverse engineering may seem to be about recovering the intentions of the original designer, but its potential is far more expansive. As the devices themselves have little regard for the intention of the designers, so too do the results of reverse engineering. A deep understanding of an artifact and its hierarchy of subsystems, assemblies, subassemblies, and components can potentially reveal new use cases, possible augmentations of existing functionality, or ways of circumventing security or safety mechanisms. In short, reverse engineering proliferates the possibilities for a device and/or a design, from maintenance to duplication to sabotage. The practice invokes a set of critical questions such as: How does the thing do what it claims? What else might it be doing or be capable of doing? Who can make it do these things? How can we fix it? How can we break it?

Beginning with the premise that "a complex hardware system can be characterized as a hierarchical structure," Rekoff presents reverse engineering as fundamentally about the discovery of "the internal particulars of that level [of the system hierarchy] in term of the things that make up the immediately subordinate level and the interconnections between those things."[47] Possible types of and relationships between "items" and "elements" are specific to the type of technology under investigation. The practitioner constantly poses hypotheses about superior-subordinate (item-element) relationships in the system, testing these hypotheses through disassembly or other means of testing. This process can begin anywhere in the system hierarchy "if one has enough information about the properties of the immediately superior level."[48] Eventually, an understanding the elements of each item emerges, including their manner of interconnection, flows of information, energy, or materials, and their functionality. And as each item "is specified in terms of the configuration of its elements, the interfaces between its elements, and the functions performed by these elements," the mechanism of operation for the system as a whole becomes apparent.[49]

In the domain of software, practitioners of reverse engineering commonly ask questions about modules, flow, control, data items, objects, and processes[50] in an effort to "analyze a subject system to create representations of the system at a higher level of abstraction."[51] But how do we articulate technical representations and functionality with questions about the nature of

---

(1985): 244.
[46] Ibid.
[47] Rekoff, "Reverse engineering," 249.
[48] Ibid.
[49] Ibid., 250.
[50] Ted J. Biggerstaff, "Design Recovery for Maintenance and Reuse," *Computer* 22, no.7 (1989): 36-49.
[51] E.J. Chikofsky and J.H. Cross, "Reverse engineering and design recovery: A taxonomy," *IEEE Software* 7, no. 1 (1990): 13, as quoted in Wego Wang, *Reverse Engineering: Technology of Reinvention* (Boca Raton: CRC Press, 2011), 18.

intelligence and data analysis and the political import of those systems? One tool for this articulation comes from new media and game studies with Ian Bogost's concept of "procedural rhetoric." For Bogost, the concept of procedurality is one of the essential properties of digital artifacts (along with participation, spatiality, and encyclopedic scope): "procedural systems generate behaviors based on rule-based models; they are machines capable of producing many outcomes, each conforming to the same overall guidelines."[52] The hinge is that "when video games represent things – anything from space demons to long-term debt – they do so through procedurality, by constructing rule-based models of their chosen topics" and these models will make different choices about where to place emphasis due to any variety of factors. In short, software objects (including games) "use procedurality to make claims about the cultural, social, or material aspects of human experience" and this is part of their network of effects.[53]

Bogost presents procedurality as a way to break the hold visual rhetoric has on studies of digital artifacts. The visual images that accompany computer processes (including video games) are subordinate to the fundamental aspect of computing machinery – that is, a rule-based system that creates a certain space of contingency and possibilities, ultimately constrained by arrangements of semiconductors. Procedural rhetoric, then, refers to the class of arguments that occur through processes (and not only representations). These "arguments are not made through the construction words or images, but through the authorship of rules of behavior, the construction of dynamic models. In computation, those rules are authored in code, through the practice of programming."[54] In other words, procedural rhetoric tends toward making "claims about *how things work*" (or don't work) in other material and/or conceptual systems.[55] Examining the procedural rhetoric of video games requires foregrounding the experience of the player as an active agent within the game environment, attending to the necessities, possibilities, and difficulties that emerge as one interacts with the artifact. Attending to the procedurality of software systems expands the task environment under consideration for reverse engineering. It challenges the analyst to specify the items and elements in terms of their contribution to both the digital functionality as well as their rhetorical functionality (what might be called in other contexts "user experience"), to situate the various modules of software systems within a higher level of abstraction, a more expansive network of effects.

But to remain at the interface with the rhetorical effects would be to ignore a key observation in Bogost's concept of procedurality, the computational contributions to these interface effects by what David Berry describes as "the performative aspects of code."[56] In the execution of compiled computer programs, the mutual implication of the material and the discursive presents itself (though not without difficulty). What Berry calls "the grammar of code" (encompassing seven ideal-types: digital data structures, digital streams, delegated code, prescriptive code, commentary code, code object, and critical code) underlies the effects of digital systems, structuring the performances in which digital computers engage in "the discretisation of the phenomenal world."[57] This grammar and computers' "internal symbolic representational structures" constitute digital systems in that they comprise the prescriptions for

---

[52] Ian Bogost, "The Rhetoric of Video Games," in *The Ecology of Games: Connecting Youth, Games, and Learning*, ed. Katie Salen (Cambridge, MA: The MIT Press, 2008), 122.
[53] Ibid., 123.
[54] Ibid., 125.
[55] Ibid., 125.
[56] David M. Berry, "A Contribution Toward a Grammar of Code," *The Fibreculture Journal* 13 (2008), accessed July 17, 2017, http://thirteen.fibreculturejournal.org/fcj-086-a-contribution-towards-a-grammar-of-code/.
[57] Ibid.

acting on these representations, the delegations of actions within and between digital systems, and other elements that are more closely linked to the technical design and engineering of computer systems and the private life of code.[58] Procedural rhetoric describes a social life of screens and interfaces, but a key interlocutor in this rhetoric context uses the grammar of code to makes its cases. And this grammar is not inconsequential, as it encodes constraints on what it means to be an object, what operations are valid, and what representations are available. With a platform like Palantir, its unique analytical grammar is a selling point, part of its value-added, but this grammar is also linked to the rhetorical and social effects at the interface and the beyond.

I see the reverse engineering as a logic that takes seriously the native, technical grammar of software objects as a way of understanding their prescribed and, perhaps more importantly, their possible operations at various levels of abstraction. Reverse engineering entails understanding the makeup and functionality of a system in its intended use case but also framing the system and its components with alternative contexts. By seeing the artifact as, for instance, partaking of a genre, we can proliferate new interactive possibilities – for repair, repurposing, replication, sabotage, resistance. Moreover, juxtaposing these technical grammars with those of cognition, knowledge, and performance opens the possibility of articulating their technical operativity with their epistemological and political generativity. Abstracting principles to from discourse and technics, the combination would speak of the digital and the political as, in Barad's terms, always implicated modes of performance, of materialization. This expanded approach augments (and complicates) reverse engineering by taking into consideration a broader task environment, a broader scene of effects. With such a juxtaposition, I hope to create one instance of what Donna Haraway called "an infidel heteroglossia" that comprehends and also goes beyond the instructions provided for the use and understanding of a political technology like Palantir.[59]

Symbiosis, Anomaly, and Crisis

The deployments of anomaly detection systems evoke a sense of urgency with respect to organizations need to recognize relevant threats and opportunities, an urgency that remains evident in Palantir's use cases and the vision for its platform. Licklider, the avowed precedent for their design philosophy, anticipates this role for computer systems in moments of crisis and situates the practical benefits of symbiosis in the contexts of high-level decision making by corporate presidents and military commanders; the Cold War military commander, in particular, must make these "critical decisions in short intervals of time."[60] This understanding of computer systems as aids to decision-making in times of crisis forms a significant part of their contemporary value in almost all sectors, from law enforcement to finance to health care. "Real-time" computer systems, originating in part with Cold War fears of a sudden nuclear strike, shrink the interval between decisions while maintaining or improving their quality in an expanding number of applications. But in this pursuit of real time, "media operate within a general structure of disavowal" according to Tung-Hui Hu, "suppressing zero time to produce a feeling of liveness (TV) or interactivity (the Web)."[61] This mediated "real time" is a fantasy

[58] Ibid.
[59] Donna Haraway, *Simians, Cyborgs, and Women: The Reinvention of Nature* (New York: Routledge, 1991), 181.
[60] Licklider, "Man-Computer Symbiosis," 10.
[61] Tung-Hui Hu, "Real time/zero time," *Discourse* 34, no. 2 (2012): 172.

crafted in response to the anxiety surrounding the "zero time" of atomic detonation, and "[w]e see fullness and presence in these imaginative or virtual reconstructions of the real."[62] Wendy Chun sees this propensity for presence and interactivity as a common element of digital systems, immersing users in experiences that demand they move quickly from information to decision, producing in users a sense of actionable intelligence. For Chun, new media are "crisis machines," and the activity of digital computing has come to be fundamentally linked its ability to churn information with "relevance to an ongoing decision."[63] Crisis here is distinct from catastrophe, in that digital systems have the (presumed) ability to transform uncertainty into recognizable and manageable risk. But in Chun's reading, "[t]he decisions we make [...] seem to prolong crises rather than end them, trapping us in a never advancing present," and we do not somehow buy a calmer future but rather produce a continuous state of emergency, "moments of fear and terror from which we want to be saved via corporate, governmental, or technological intermediaries."[64]

Here Chun turns to the role of code not only in the production of crises in new media but in the (impossible) decisions that must be made in response to the state of exception. Licklider's linking of man-computer symbiosis with the prospect of "the ten-minute war" presages the wider hope from major organizations (and we who look to be saved by them) that the movements from data to decision and finally to action can be continually reduced, that organizations can realize, at scale, the "automatic compliance" that "welds together script and force."[65] Anomaly detection systems, in their ambition to rapidly locate and elucidate moments of crisis, moments where decision is needed, reflect this potential in mundane ways – a bank automatically suspending a credit card due to anomalous purchases may elicit relief or frustration. This welding is more than a manner of speaking or user experience, as it creates circuits of efficacy and legitimacy with urgent political stakes. For instance, these shrinking decision intervals are cited by a United States Department of Homeland Security as a major value in their use of Palantir platform, with its ability to capture, contextualize, and circulate evidence between field agents (using mobile devices), analysts, prosecutors, and judges, ultimately accelerating the obtainment and execution of arrest and search warrants.[66] In effect, Palantir provides a technical system that enables law enforcement to bring its judicial supervision "in house," as it were, diminishing the interval between police conception of an act and its execution. As this interval shrinks, we see the ways in which digital systems become part of the very being of police action and the security apparatus, not just a convenient medium for their communications. This returns us to the ontological effects of computing machinery and, in particular, to the rhythms and timings that attend it. What intervals, what timings does "security" demand? Due process? Democracy?

Though Chun's analysis proceeds primarily through Agamben's conceptions of sovereignty and the state of exception, the list of newly empowered "saviors" includes corporate and technological intermediaries alongside government, suggesting a fragmentation and dispersion of authority from its properly political sites. One striking characteristic of the Palantir platform (and similar offerings from other companies) is the wide variety of institutions it is claimed to be designed for, the claim that all organizations, be they government or enterprise,

---

[62] Ibid.
[63] Wendy H.K. Chun, "Crisis, crisis, crisis, or sovereignty and networks," *Theory, Culture & Society* 28, no. 6 (2011): 96.
[64] Ibid., 97, 99.
[65] Ibid., 102.
[66] Palantir Technologies, "Building a Human Trafficking Case from a Lead to Arrests," YouTube video, 20:20, posted by "Palantir," July 15, 2013, https://www.youtube.com/watch?v=gS1IMB--3dw.

share a common need for tools of risk analysis and anomaly detection – or crisis production and management – the onus of recognizing and responding to threats, the infrastructure for doing so is certainly no longer limited to the state. The infrastructures of crisis are increasingly dispersed. The construction of such systems may signal a shift away from the nation-state as the center of technopolitical critique, as a national government or agency becomes one customer and/or vendor in a larger field of non-democratic firms and organizations exchanging access to datasets, analysis tools, and other technologies of modern governance. If the problem of the state is partly defined by its "prism" of reflective practice, the objects and logics of its intervention, we should look critically at when and how those practices and those technologies find their ways into different institutional contexts with different requirements for accountability and legitimacy.

Chapter Outline

Palantir advertises deployment of two versions of its analysis platform, "Gotham" and "Metropolis," and while the two share common interface characteristics and datastore technologies, each is centered on a different core model for the analyses it hosts. Palantir Gotham (previously referred to as Palantir Government) was first designed for intelligence agencies, law enforcement, and cybersecurity firms and is centered on the object graph. This approach models a domain as populated with objects connected to each other by relationships, which can be presented as now-familiar network visualizations. As such, Gotham works with what is called relational and/or transactional data that indicate connection and interaction between discrete entities: transfers of money between bank accounts; e-mails and phone calls exchanged between individuals; kinship and organizational membership. Gotham presents *any and all* of these kinds of data on a single graph within the environment. Palantir Metropolis (previously called Palantir Finance) was designed with investment banks and hedge funds in mind, and it centers on the time series as the core data model. With tools to visualize, manipulate, and analyze series of quantitative values (e.g. asset prices), the Metropolis user examines charts, performs regressions, and tests strategies in an effort to discover patterns in the movement of markets and new opportunities for profitable investment.

Chapter One centers on a close reading of company-recorded demonstration video (via YouTube) to elucidate the investigative rhythms and knowledge practices implicit in Palantir Gotham. Gotham, geared toward law enforcement, intelligence, and other security professionals, supports rapid, interactive analysis of non-quantitative data across multiple heterogeneous datasets. These investigations rely primarily on graph visualizations and aim to uncover anomalous (elusive and/or illicit) network structures not immediately obvious to unaided human *or* machinic detection systems. The results are investigative narratives that name, explain, and target instances of elusive, anomalous, or threatening behavior mediated by digital traces. These structuring categories, however, are neither fundamental nor disinterested but emerge from highly cathected figures of organizational risk including whistleblowers, malicious insiders, cyber criminals, and terrorists. I argue that such data-based, organizational "intelligence" is a constructed effect of distributed cognitive assemblages and that a variety of agents and epistemic mediators – from machine learning algorithms to graph visualizations, to experienced human analysts – shape the resulting possibility space, the concepts invoked, and the analysis produced.

Chapter Two addresses the ontological suppositions embedded in the database design of the Palantir platform and traces these contributions of the "deeper" machinic processes that render heterogeneous digital data sets as “objects” (e.g. entities, events, and documents) in the unified visual environment. The implications of knowledge producing systems like Palantir derive not only from the spectacular “front end” moments of human-computer interaction, but also from the automatic, machinic configurations that collect, store, and organize these data on the “back end.” This examination primarily emerges from readings of Palantir's patent portfolio, training and demonstration videos, and related literature describing the functionality of comparable database systems. The chapter examines of three successive layers of abstraction in the Palantir data platform: the distributed datastore, the key-value driver, and the dynamic ontology. I describe how each of these computational subsystems contributes to the ‘cutting out’ of discrete objects (described by organizationally defined ontologies) from data streams to provide source material for subsequent narrative sutures. Database ontologies serve as semantic ‘hooks’ for data analysis and interpretation, supporting the assembly of a narrative with Palantir objects serving as agential pivots for sequences of events. I argue that the implicit design objective for Palantir’s object model and dynamic ontology, with respect to techno-epistemology, is a machine whose activity can quickly and readily contribute to the semantic strata of investigation and analysis, an objective partially achieved through the ‘packaging’ of digital traces as typed data objects.

Chapter Three turns to Palantir Metropolis, the version of their platform for domains primarily characterized by quantitative and time series data and targeted at financial firms looking to understand market dynamics and create novel trading strategies. The chapter draws from a procedural analysis of the user interface and prototypical workflows. With respect to the common theme of *anomaly*, Metropolis presents a contrast to Gotham in that market anomalies are often pursued as opportunities rather than threats. In examining these epistemic performances, from human clicks to underlying data structures, my goal is to provide an account of how Metropolis frames the discovery or construction of trading strategies and opportunities in financial markets. Traders are engaged in what I call “competitive insight,” attempting to uncover and exploit discrepancies between the market price and the “true” price of assets before others notice, and Metropolis is just one of a number of tools for such competition. The performativity of economic theory and trading practices has been the focus of a number recent studies in the economic sociology and the sociology of markets, and the effect of intensely computerized systems like algorithmic and high-frequency trading has garnered increasing interest. A product like Metropolis, reliant as it is on human timings, seems to represent a more deliberate approach to distinguishing the apparent from the real in capital markets and suggests that the consequences of slow finance may remain significant. Derivatives and the digitization of finance introduce new layers of abstraction for value and an increasing sense of virtuality into the economic. My examination of Palantir Metropolis aims to elaborate the design of this hybrid cognitive assemblage, one that makes sense of this abstraction and virtuality, with the understanding that such sense making plays a role in constituting the financial reality it claims to represent.

Chapter 1
Intelligence in the Break: Discovering Networks and Rethinking the Interface

1. A Palantir workflow: data presented "the way the world really works"

Palantir's government-focused offering, Gotham, is more than just an isolated software product for data analysis. Rather, it is a platform or possibility space directed toward interactive data analysis and an active component of what I am calling a larger epistemic performance. At the center of many of Palantir's recorded demonstrations is the placing of the viewer beside an engineer as they go through a "workflow," an exemplary process of turning computerized data sets into some analytical knowledge. The motivating question behind many of these investigation amounts to "Why are we seeing this pattern? What does this mean?" The process, dramatized in these manipulations of interface artifacts along with a "thinking out loud" narrative from the demonstrator, suggest a kind of conversation between the human analyst and the software environment. Drawing from Ian Bogost's work on the rhetoric of video games, we might say that the procedural rhetoric in Palantir's Gotham, the programmed practices of interaction, puts forward a particular ideology of composite human-computer systems engaged in investigation, insight, and intelligence.[1] Though these investigations are presented as instances of detection or discovery of pre-existing phenomena, focusing attention on the patterns of interaction between analyst and artifact opens a discussion as to the constructive character of digital data analysis as moments of guided construction rather than sudden revealing.

With respect to software objects like those made by Palantir, we can begin by attending to two sets of features. First, we can look at the interface as a dynamic set of interactive elements, forming part of the limits and scope of what is possible within the software environment. Second, the training materials and demonstrations from Palantir foreground what they see as the optimal patterns of interaction with their environment and provide a sense of what emerges from those optimal workflows – the processes and results that form the value-added of the product. In this chapter, I will follow along with one of these demonstrations, pitched as prototypical use cases for these software objects, and look over the shoulder of the analyst. Taking a cue from Palantir's discourse, I highlight the investigative quality of the demonstration, with its connotations of hypotheses, exploration, discovery, testing, and revision. The investigative narratives in these demonstrations reveal emphases in the Palantir Gotham environment as to what the key components of a successful investigation are, their sequence and logic, and their visualization.

A presentation in the summer of 2013 at the conference "Palantir Sync: Investigate" entitled "Building a Human Trafficking Case from a Lead to Arrests," features an official from the U.S. Department of Homeland Security describing their efforts to deploy the Palantir platform.[2] The official highlights DHS agents' enthusiasm for the rapid searches now possible from mobile devices and the ability to quickly interface with command staff and the United States Attorney's office. Following this introduction, a Palantir engineer guides the audience through an operation in New York. The data has been notionalized for privacy reasons, but,

---

[1] Ian Bogost, "The Rhetoric of Video Games," in *The Ecology of Games: Connecting Youth, Games, and Learning*, ed. Katie Salen (Cambridge, MA: The MIT Press, 2008).
[2] Palantir Technologies, "Building a Human Trafficking Case from a Lead to Arrests," YouTube video, 20:20, posted by "Palantir," July 15, 2013, https://www.youtube.com/watch?v=gS1IMB--3dw. Further quotes in this section refer to this presentation unless otherwise indicated.

according to the engineer, the presentation is based on a real case and “workflow.” In introducing the Palantir Gotham interface, the engineer describes the different applications (accessible via buttons at the top of the window, below the menu bar) as different “rooms for performing analysis”: a map room, a graph room, an object explorer, and a browser.

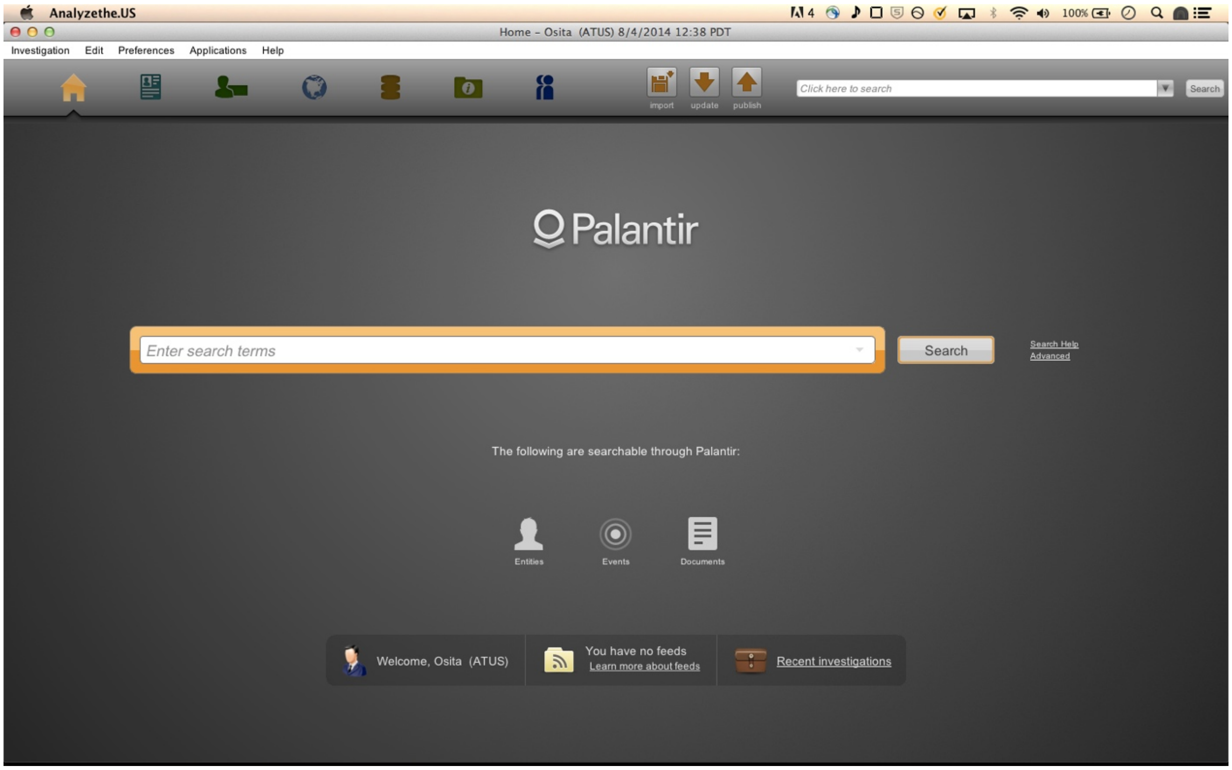

Figure 1.1. The Palantir Gotham application's home tab. Author screenshot.

The Graph application "provides a way to visually explore the semantic relationships between data objects [...] represented visually as networks of nodes and edges,"[3] while the Map application includes features like heat map visualization. The Object Explorer “allows users to drill down on objects of interest within massive datasets,” while the Browser is a space for users to view raw text documents and other unstructured data, allowing them to tag elements to make them more easily usable and searchable for investigations.[4]

[3] “Palantir Gotham: Applications,” Palantir Technologies, accessed September 30, 2014, https://www.palantir.com/palantir-gotham/applications/.
[4] Ibid.

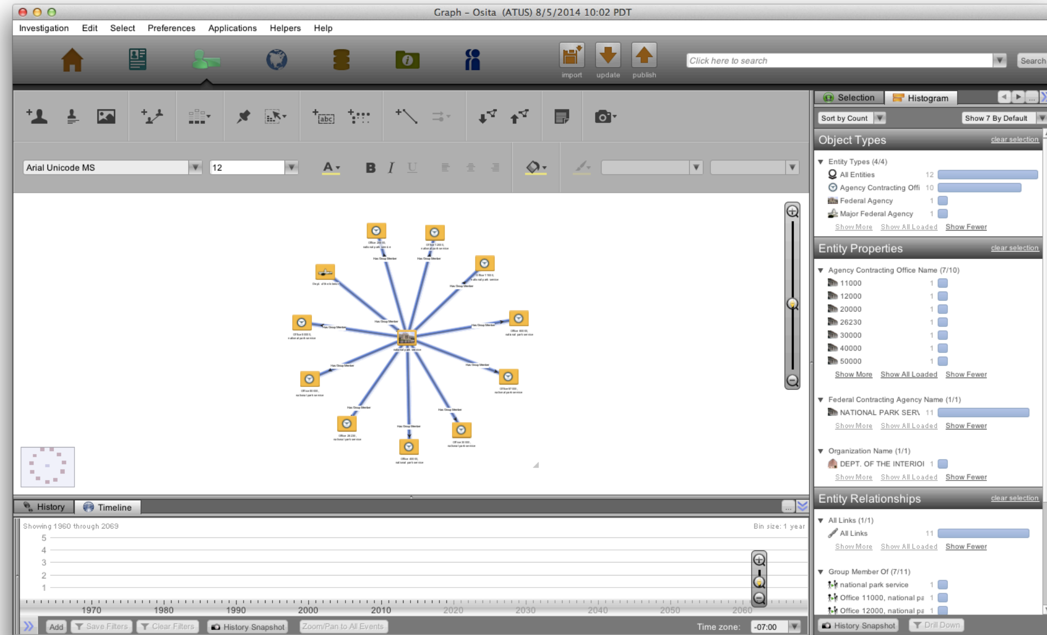

Figure 1.2. The Graph application in Palantir Gotham. Author screenshot.

Figure 1.3. A larger example of a social network analysis graph. By Martin Grandjean.[5]

The presenting engineer then turns to the lead that begins the investigation, an alert from a bank along with transaction records in familiar spreadsheet format. The engineer takes issue with this format for investigation, as it is "difficult to find common threads." Luckily, Palantir

[5] Martin Grandjean, "A force-based network visualization," image, November 2, 2013, https://en.wikipedia.org/wiki/Graph_drawing#/media/File:Social_Network_Analysis_Visualization.png. Creative Commons License (CC BY-SA 3.0), http://creativecommons.org/licenses/by-sa/3.0.

can import this spreadsheet and model the data therein using its own logic, with the engineer claiming, “We model data the way human beings think about the world.” Bringing the data into the Palantir Gotham environment, we see the engineer give us a representation of this information on the Graph, where we can “see human beings linked to each other through bank transactions or phone calls – the way the world really works.” A visualization of nodes and edges connecting the senders and recipients of the transactions from the spreadsheets appears in the graph application. The engineer then opens the “flows” helper, which provides an animated visualization of the bank transfers – a red orb moving along the line connecting two persons, its size proportional to the amount.

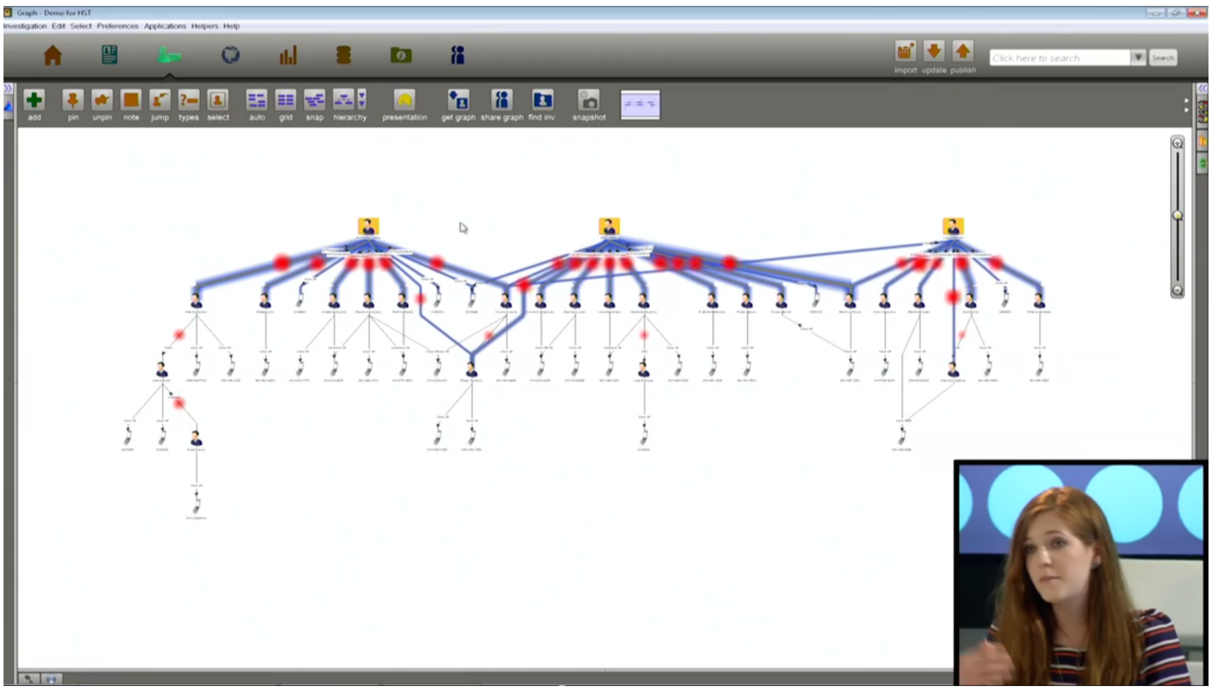

Figure 1.4. Bank transactions in the Graph application with the "flows" helper activated from the Palantir demonstration, “Building a Human Trafficking Case from a Lead to Arrests.” Author screenshot.

Following this, the audience is informed that “money generally doesn’t flow in such structured and organized ways.” At this moment, it is the human analyst noticing structure where it should not be, an anomalous moment of organization, that tips them off to potentially illicit activity. After the analyst or investigator has “noticed” this anomaly, they may (as the demonstrating engineer does) perform a search against their existing databases to see if there are names or addresses in common. If names or addresses from the third party (bank) information match the database, the analyst can then perform a “search around” for further linked entities and documents (as well as events, the other “first-order” object type in Palantir Gotham). This act of “searching around” is pivotal in most of the investigative narratives Palantir presents. It returns these linked objects to the graph, contributing to the sense that the analyst is, in stepwise fashion, uncovering various elements and aspects of a network or organization. The graph is also a place where this visual sense of organization can be arranged, auto-arranged, reformatted (nodes no longer under consideration can be removed), tagged and color-coded, or animated (with the flows helper, what the engineer off-handedly refers to as the “hypnotics”).

The demonstrating engineer then turns to the map application, wanting to see if there's any interesting information related to an address from the bank transactions that appeared in the database. To do so, the engineer simply drags the "person" from the graph application onto the map and performs a radius search to see if there are, for example, any tip line reports in the area that would suggest human trafficking. For the sake of brevity, the engineer tells us that it would then make sense to subpoena phone records for the cell phone in the database connected to Irma Mendez, and then adds this work by a collaborator to her existing graph. Two of the numbers are associated with persons already in the structure gleaned from the original bank information, and there is a temporal pattern to the phone calls (Mondays between 10 a.m. and 11 a.m.), leading our analyst to believe that those two women may be victims of human trafficking (making calls to handlers and payments). Part of Palantir's emphasis is that links like this, of different types and from different data sets, coexist on the same graph, as real-world connections are often multimodal. The engineer ends with a verbal summary of the demonstration highlighting: the conversion of the bank's information from an "unmanageable format" into a graphical representation; the ability to then search against their existing data sets for name and address matches; geosearch capabilities to find tip line reports around an address of interest; and the ability to do temporal pattern analysis of phone calls. But the major value added, presumably, is that all of this is possible within the same environment, that different "rooms" for analysis are all brought onto a single screen and a user can move information to and from.

The Homeland Security official returns to the stage and reminds the audience of the features that he, as a law enforcement agent, has found to be most useful. He first remarks on the rapid communication allowed by the Palantir Mobile application – with a version of the platform installed on agents' phones, text messages and pictures allow supervisors to coordinate operations and raids, as well as to communicate with federal prosecutors regarding potential evidence, who in turn can rapidly communicate with judges regarding warrants. The final critical feature the DHS official points to is the ability to not just quickly search large data sets, but to rapidly visualize potential organizational structures for such criminal activity, based on cash flows, phone calls, or other information. According to this law enforcement official, the graph is not just a convenient representation, but also a critical part of the workflow and the value-added of the Palantir Gotham environment.

Based on this demonstration, what can be said about Palantir Gotham's procedural rhetoric? The applications (and the engineer's description) make a claim about different salient aspects of investigation and analysis – there are semantic relationships explored on the graph, spatial relationships explored on the map, and documents and objects waiting to be brought into these one or both of these primary "rooms" for analysis, with the graph being the point of emphasis. The engineer foregrounds the movement of objects back and forth within and between the applications, from Map to Graph and back again, as the way that one fleshes out investigative hypotheses and evaluates them. In particular, the activity of "searching around" for other data objects linked to an entity or entities leads to a moment where the analyst "notices" something exceptional or anomalous. In turning away from the spreadsheet as a site of analysis, the Gotham platform engages (or claims to engage) human intuitions surrounding organization in a visual environment. But this engagement is imprecise, iterative, and, in a word, messy. Precision instruments, whether in the spreadsheet structure or with digital calculation, play a surprisingly minor role in this arrangement. We are left to wonder: on what terms could this be called the future of analysis in the age of "Big Data"?

2. *Friction, or, "It's not because we have faster computers…"*

Along with demonstrating some specific features of the Palantir platform, the human trafficking demonstration introduces the company's emphasis on human-computer interaction (HCI), primarily interface and data abstraction. This emphasis on HCI, on the "mesh" between human users and computer machinery, is the explicit focus of two talks given by Palantir engineers at other conferences organized by the company.[6] Both talks begin with discussions not of law enforcement or intelligence analysis, but of chess. Chess is seen as a particularly useful laboratory for studying human and machinic problem solving, given established methods for ranking players and, at present, a history of purpose-built chess computers that routinely compete with human grandmasters. Specifically, the talks open with a provocative story from the PAL/CSS freestyle chess tournament of 2005. In this tournament, an IBM purpose-built chess computer named "Hydra" was pitted against human grandmasters aided by laptops, as well as a field of amateur players. The computer-aided grandmaster defeated Hydra, the explanation being that the grandmaster's experience and intuition allowed them to narrow the problem-space at a given point in the game, whereas Hydra's "brute force" approach (as a deterministic computing machine) must try and evaluate all possibilities.[7] The surprise came when the grandmaster faced an online amateur with the handle "ZackS" (sometimes written as "ZakS") – ZackS soundly defeated both Hydra and the grandmaster. On further investigation, ZackS turned out to be two men in New Hampshire (Zackary Stephen and Steven Cramton), using three consumer computers running four different chess engines.[8] Steven Cramton gives a description of their process in a 2005 interview for the website *ChessBase*:

> I think that we have always had a slightly different approach to chess than most others and we are not afraid to condemn certain respected positions after extensive analysis and we often find ourselves searching for the truth of many different openings.[9]

Russian grandmaster Garry Kasparov, writing about the tournament for the *New York Review of Books*, gave this analysis of the result: "Weak human + machine + better process was superior to a strong computer alone and, more remarkably, superior to a strong human + machine + inferior process."[10] Cramton describes this "better process" in a bit more detail:

> Candidate moves were usually chosen based on our own experience in certain types of positions or by the consensus of the computer [chess] engine programs. Once we established our possible candidate moves (usually three or less, but sometimes more) we began to investigate the lines extensively. Zack would analyze a few lines and I would

---

[6] Palantir Technologies, "Human-Computer Symbiosis: Efficient and Powerful Use of Computing Power," YouTube video, 37:43, posted by "Palantir," November 3, 2010, https://www.youtube.com/watch?v=oLalkcMDCwg; Palantir Technologies, "Friction in the Machine: How Fluid Processes Allow Optimal Human-Computer Interaction," YouTube video, 38:37, posted by "Palantir," November 4, 2011, https://www.youtube.com/watch?v=Kw1hZkfOVhQ.

[7] Palantir Technologies, "Human-Computer Symbiosis" (video).

[8] "PAL / CSS report from the dark horse's mouth," *ChessBase*, June 22, 2005, http://en.chessbase.com/post/pal-c-report-from-the-dark-horse-s-mouth.

[9] Ibid.

[10] Garry Kasparov, "The Chess Master and the Computer," *The New York Review of Books*, February 11, 2010, http://www.nybooks.com/articles/2010/02/11/the-chess-master-and-the-computer/.

> analyze a couple of different lines. When either of us found a strong continuation we then looked at it together, comparing the lines between the different [chess] engines (mainly between Shredder 8 and Fritz 8).[11]

The Palantir speakers place the ZackS story and Kasparov's insight at the center of the Palantir design philosophy, with one presenting engineer emphasizing that the "magic" is in the process, the interaction between humans and machines (this engineer offhandedly refers to the "crazy ballet" that must have taken place in ZackS's room).[12] They see the purpose of Palantir, as a digital artifact, as facilitating this improved process for human-computer problem-solving.

ZackS's victory recalls a number of debates about the limitations of computers and, in particular, the continued disappointments in the field of artificial intelligence since the heady days of the 1950s computing pioneers. Computability and complexity theory, a subfield of computer science, classifies computing problems based on their tractability for certain kinds of systems, and the speakers give us a rough set of categories for understanding these failures in what we now call "artificial general intelligence." Returning to chess, the problem space in this case is too big – in the other Palantir engineer's talk on this topic, the speaker calls our attention to the Shannon number, information theorist Claude Shannon's estimation of the number of possible board arrangements in chess: 10^{43}, a number that dwarfs estimates of the total available data storage on earth (about 10^{21}).[13] And this may even be a low estimate – that engineer mentions Victor Alice's estimate (based on the game tree complexity) of 10^{123}! Even in the 21st century, with computing power growing exponentially, humans keep pace with and often defeat chess computers. In other cases, the problem space may be too dynamic. One Palantir presenter points to cyber security and the difficulty computers have in detecting network intrusion. Human adversaries are adaptive, and, in comparison to chess, there are relatively few "rules" for what actions are possible or permissible. Even statistical machine learning models fall short, as human hackers consciously change tactics to hide from detection techniques based on data from past attacks. Finally, there are problems that involve too many domains – that is, much of social reality. In these three cases, humans seem to outperform the best digital computers.[14]

As a response, Palantir's design philosophy centers on the idea that, for certain problems, computers should primarily serve as a kind of game or play space, a possibility space. The analyst uses data to generate the graph representation, allowing her to see the "flows" of money and the connections between people. She manipulates, arranges the icons to make the connections clearer and more distinct to her, searches for other, related entities and decides whether or not their presence in the graph makes sense (and what kind of sense), deletes others, rearranges, and calls on information from colleagues. And this does not happen through the use of some formal logic or procedure, but emerges in the process of clicking, dragging, looking, adding, deleting, looking again. The computer performs the clerical work of searching and preventing simple mistakes, providing a space for rapid visualization on screen – imagine trying to draw out one of the object graphs by hand! – and reducing the latency of other routine tasks.

To drive home this claim, one "Friction" presenter ends his talk with an example from the financial sector, a brief story about analysts using the Palantir platform to explore the

---

[11] "PAL / CSS report from the dark horse's mouth," *ChessBase*.
[12] Palantir Technologies, "Human-Computer Symbiosis" (video).
[13] Palantir Technologies, "Friction in the Machine" (video).
[14] Palantir Technologies, "Human-Computer Symbiosis" (video).

performance of the S&P 500 as an investment vehicle in comparison to a few alternatives.[15] The thrust of the argument is that this workflow – which involved creating a number of hypothetical indices weighting component stocks differently and comparing their performance to the original index – took about five minutes with the Palantir platform, while with other, current platforms on Wall Street it would take three days. "And it's not because we have faster computers," the speaker says, rather it is that they have "*arranged all the pieces in such a way that there's no friction*." [emphasis mine] The subsequent claim is that the Palantir platform "deconstructs" data analysis procedures and optimally reconfigures its elements, providing analysts with various basic actions that can be composed and recomposed while leaving the overall direction of analysis to the human in a way that ends up being "much more expressive against the problem."

If your goal is more efficient and accurate problem-solving, if you want to improve what the Palantir speakers call "analytical capability," you need to focus on the whole system, or what J.C.R. Licklider called "man-computer symbiosis."[16] The man-machine systems Palantir puts forward as inspirations are framed in terms of this symbiotic understanding of computation, intellectual activity, and decision making. Palantir's design philosophy references Licklider's work explicitly on a number of occasions, using his notions of symbiosis and friction as a way to understand intellectual activity.

$$\mathrm{a}(c, f) = H \frac{c}{(1+f)}$$

Figure 1.5 Palantir writes the revised function describing analytical capability; In this equation, H represents the human's intellectual capability, and c represents the computing power of the system. The parameter f, friction, is analogous to the coefficient of friction in physical systems and usually is somewhere between zero and 1.[17]

This design philosophy prominently features a human role in problem-solving systems; the approach emphasizes process, "friction," as the dominant parameter affecting that system's analytical capability. In particular, *process* refers to the "clerical and mechanical" activities that impinge on the time the man spends on the "intellectual" aspects of the problem.[18] This recalls the common erasure of women's contributions to the history of computing and computation and reinscribes their presumed passivity as mere calculators, receptive to the commands of the male intellectual figure who does the real thinking.[19] This rendering of Licklider, however, tempts us to flatten the landscape of thinking and problem-solving, with the implication that the systems and design approaches that bring success in a narrow set of idealized domains (e.g. choosing a move in a chess game) will bring similar success across a range of domains. Presenting friction

---

[15] Ibid.

[16] J.C.R. Licklider, "Man-Computer Symbiosis," *IRE Transactions on Human Factors in Electronics* 1 (1960): 4-11.

[17] Ari Gesher, "Friction in Human-Computer Symbiosis," Palantir Technologies, March 8, 2010. Accessed May 9, 2016, https://palantir.com/2010/03/friction-in-human-computer-symbiosis-kasparov-on-chess. Page removed, archived version from July 2, 2016: https://web.archive.org/web/20160702013739/https://www.palantir.com/2010/03/friction-in-human-computer-symbiosis-kasparov-on-chess/.

[18] Licklider, "Man-Computer Symbiosis," 6.

[19] We see this in Licklider's discussion the decisions of military commanders and corporate presidents as the exemplary use cases for such technologies.

as a scalar value elides the ways in which the machine symbiont presents problems and defines permissible solutions – that is, there will be differences not only of quantity with respect to the effectiveness of a symbiotic arrangement but also with respect to quality – even between systems ostensibly oriented toward the same problem. While I take up this critique of Licklider in more detail at the conclusion of this chapter and at the beginning of chapter 2 of this dissertation, I briefly introduce his thought here to frame the two sections that follow. The first addresses a second example from Palantir's design discourse, that of protein structure prediction, and the second explores the concept of problem structure and its impact on the design problem-solving systems. Through the course of both, I raise questions regarding the quality of the symbiotic interactions between human and machine. While "conceding dominance in the distant future of cerebration to machines alone," Licklider turns our attention to the long (perhaps indefinite) "interim during which the main intellectual advances will be made by men and computers working together in intimate association."[20] What, then, are some of the concrete contributions of humans and machines in these problem-solving situations? Moreover, how is the "real" thinking of the "man" in the "man computer symbiosis" potentially shaped by the computer that, supposedly, merely receives and represents his commands and their results?

[20] Licklider, "Man-Computer Symbiosis," 5.

3. Symbiosis (1): Finding Protein Structures

The other problem-solving competition discussed in both of the Palantir “Friction” presentations is protein folding, a notoriously difficult computational task. Despite the seeming simplicity – proteins are linear strings of a limited number of residues (amino acids), predicting their three-dimensional conformation with computers has not yet been successful. In 2008, researchers at the University of Washington released the puzzle “game” called FoldIt, which challenged human players to find the three-dimensional conformation of proteins.[21] Their solutions were evaluated based on the energy state (protein folding tends to follow the “path of least resistance” in settling into the lowest possible energy state) and compared to solutions given by the most advanced computational approaches to the folding problem.

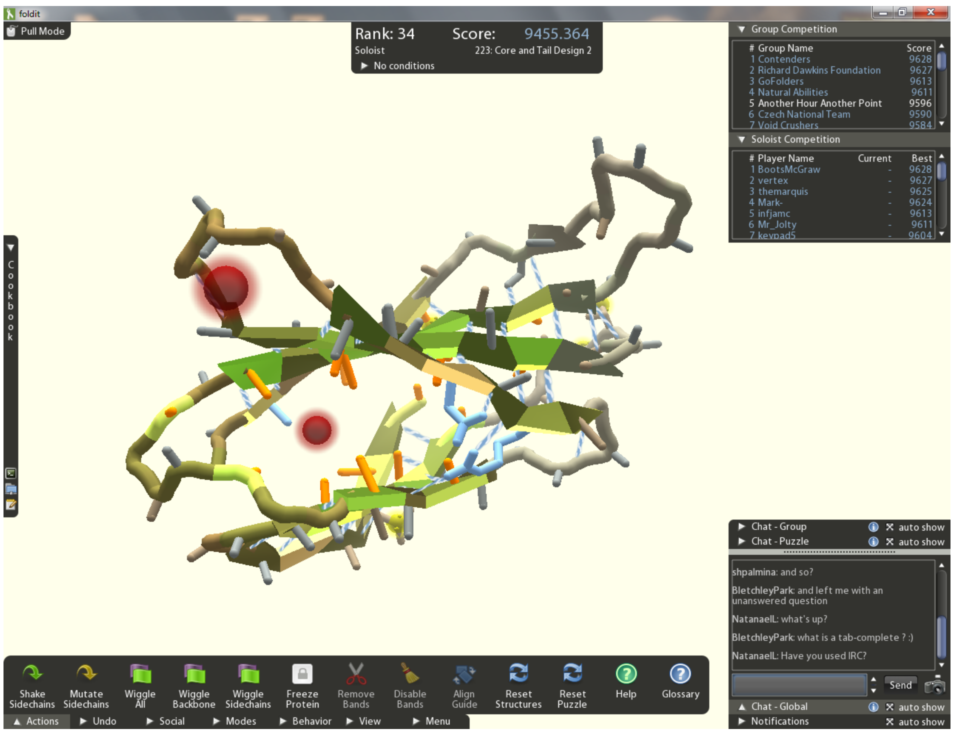

Figure 1.6. Screenshot from the “FoldIt” application. The interface includes the “moves” for reconfiguring the protein conformation (near the bottom), as well as a group chat window for collaboration (lower right corner).[22]

In 2010, the researchers published results showing that human “players” had found conformations that either matched or were significantly better than those found algorithmically. This result reignites discussion about what kinds of problems are solved more efficiently by humans or human-directed computation, but what the Palantir speakers do not emphasize is the FoldIt designers’ description of the tool as a puzzle *game*. In FoldIt, the players “interact with protein structures using direct manipulation tools and user-friendly versions of algorithms from the Rosetta structure prediction methodology.”[23] Example “moves” include: “combinatorial side-

[21] Seth Cooper et al., “Predicting protein structures with a multiplayer online game,” *Nature* 466 (2010): 756-760.
[22] Animation Research Labs, University of Washington, “Screenshot von FoldIt,” image, November 16, 2009, https://commons.wikimedia.org/wiki/File:Foldit_screenshot.png. Creative Commons License (CC BY-SA 3.0 Germany), https://creativecommons.org/licenses/by-sa/3.0/de/deed.en.
[23] Cooper et al., “Predicting protein structures,” 756.

chain rotamer packing ('shake'), gradient-based minimization ('wiggle'), fragment insertion ('rebuild')."[24] In other terms, FoldIt players click, drag, and above all *play* with proteins, on screen, to find their optimal conformations, and in doing so outperform computers working alone.

Purely automatic approaches to *de novo* protein structure prediction face (and have addressed, at least partially) a number of challenges.[25] As noted above, FoldIt was designed as an extension of the protein structure prediction program Rosetta in an effort to address a number of persistent limitations to that purely mechanical approach to the problem. We might understand FoldIt as injecting certain human performances into a largely algorithmic problem-solving process for protein structure prediction – in a strange inversion, the human augments the computer. Developed by the Baker laboratory at the University of Washington, Rosetta approaches the problem of protein structure prediction using a two-stage approach, with the first, ab initio stage being "a coarse grained fragment-based search through conformational space… that favors protein like features," and the second "relax" stage is an "all-atom refinement using the Rosetta full-atom forcefield."[26] In particular, I want to highlight what researchers see as the source of the specific challenges to locating a solution in this problem space, that is, finding a global free energy minimum amongst a huge number of possible geometries. Anfinsen outlines a number of physical factors that govern the folding of protein chains — that is, their assumption of a three-dimensional geometry with the lowest free energy (what Anfinsen calls the thermodynamic hypothesis).[27] Deriving this lowest energy conformation from the primary structure (the linear sequence of constitutive amino acids in the protein) for the bacterial ribonuclease described in the paper (149 residues) would require calculating and comparing the free energies of between 4^{149} to 9^{149} conformations.[28] As in the case of chess and the Shannon number, this conformational space far exceeds current computational capabilities, making a purely brute force approach infeasible at present.

Evidence suggests that protein folding takes place in phases, with a relatively small number of low energy "pathways" presenting themselves as the only feasible candidates for the start of the folding process.[29] Thus, the range of possible solutions may be narrowed using hypotheses about nucleation events and subsequent cooperative stabilizations – that is, certain sequences of residues (or fragments) in a protein tend to assume certain shapes, and this can be the basis for discovering the optimal conformation. In other words, folding is not a globally spontaneous moment for the protein; rather, different domains of the macromolecule take shape over time. Informed by these hypotheses, Rosetta implements a heuristic search of the problem space with a combination of stochastic and deterministic algorithms. In effect, Rosetta models not just the energetics but the kinematics of folding, presuming that protein folding occurs in

---

[24] Ibid.
[25] *De novo* protein structure prediction generally refers to attempts to determine the optimal three-dimensional configuration for a protein based only on knowledge of the primary sequence of amino acids and without the use of experimental characterizations (e.g. x-ray crystallography).
[26] "Abinitio Relax," *Rosetta Commons*, accessed December 15, 2006, https://www.rosettacommons.org/docs/latest/application_documentation/structure_prediction/abinitio-relax.
[27] Christian B. Anfinsen, "Principles that Govern the Folding of Protein Chains," *Science* 181, no. 4096 (1973): 223-230.
[28] Ibid., 228.
[29] Ibid.

stages as pieces of the macromolecule stabilize and modify the energetic favorability for certain geometrical configurations in other regions of the macromolecule.

Based on the understanding that "folding takes place when a combination of local conformations is sampled that makes possible low-energy tertiary interactions," Rosetta breaks down this search of the conformational space into two phases – a coarse- and a fine-grained search – wherein the first "ab initio" phase samples the protein sequence in an effort to find fragments that are anticipated to form common local low-energy structures (loops, beta-sheets, alpha helices), what the authors refer to as low-resolution fragment assembly,[30] Through this stochastic "sampling" of the energy landscape, Rosetta chooses a set of low-energy candidates for the high-resolution refinement stage. In this second, computationally intensive "relax" phase, the algorithm adds back atomic detail and through stochastic and deterministic procedures further "adjusts" those candidates in an effort to discover the lowest energy structure that accounts for the interactions of all the atoms in the macromolecule.

This second "relax" (or "all atom adjustment") phase touches on a particularly challenging aspect of the process and provides some suggestion as to the potential contribution of human performances. One critical feature of the protein structure problem Rosetta's developers identify is the "ruggedness" of the energy landscape. In many cases, the low-energy conformation may be "surrounded," as it were, by extremely high energy intermediate conformations that would seem to preclude a given folding pathway as the route to the optimum.[31] In other words, sometimes intermediate states of a puzzle or problem may exhibit features that, according to the evaluation criteria, would suggest we abandon that specific approach as unlikely to yield a solution. Rosetta's sampling and adjustment procedures operate on this principle in an effort to reduce the computational resources needed to find an optimal solution. However, this perceived likelihood, based on currently observed features, is far from a foolproof guide. The contribution of the human "player" in FoldIt seems to be most significant in this second phase, where a number of candidate structures generated by Rosetta are presented to players, who choose, based on the program's highlighting of energetically "frustrated" areas in the candidate structure and previous gameplay experience, the best candidate for restructuring "by hand."[32] This work by hand involves moves where the relevant adjustments would be coordinated shifts relieving energetic frustration at multiple sites. These features of the protein folding problem space resonate with a symbiotic approach, and the effectiveness of human contributions correlate to the plethora of close-but-not-quite structures, analogous to a puzzle that seems nearly complete.

FoldIt researchers put forward two hypotheses, "[v]isual problem solving" and "strategy development," to explain the increased intellectual yield of the hybrid system, which outperforms either Rosetta or a human alone in predicting de novo protein structure.[33] First, Cooper et al. suggest that human spatial reasoning improves both the "sampling of the conformational space" (recognizing which among the number of "coarse" structures assembled from fragments with local energy minima are the best candidates for adjustment) and recognizing when a suboptimal conformation (that is, one with a higher free energy that might be dismissed by the algorithm's heuristics) may, with adjustment, ultimately provide a path to the

[30] Rhiju Das and David Baker, "Macromolecular Modeling with Rosetta," *Annual Review of Biochemistry* 77 (2008): 368.
[31] Ibid., 367.
[32] Cooper et al., "Predicting protein structures," 759.
[33] Ibid., 756.

native conformation.[34] Second, the researchers posit that the multiplayer capabilities of FoldIt, which encourage competition and collaboration, allow players to not only explore the conformational space (i.e. the energy landscape) but also to explore different strategies for exploring this space, as well as providing for division of labor amongst team members whose facility with those strategies may vary.[35] FoldIt provides a platform for not only search but collaborative meta-search of the problem space.

The features of the FoldIt game and the process by which its players, presumable amateurs with respect to the study of protein modeling, find optimal conformations has significant precedent in expert practice. Natasha Myers's ethnographic account of crystallographers highlights these professional scientists' frequent appeals to embodied intelligence in the manipulation of physical models and graphical renderings of proteins. These practitioners are enmeshed in an intimate and affective relationship with their objects and models, and their construction and manipulation of these models constitute a genre of "body experiments" that are "reflexive, improvisational, and exploratory."[36] While part of the effectiveness of FoldIt may indeed derive from its reconfiguration of the problem of search with respect to protein folding, Myers's analysis suggests that the "kinesthetic imagination" of the human players might be behind their success.[37] For an expert crystallographer, the body becomes a tool of reason and a vehicle for incorporating structural knowledge into a felt experience. The "haptic creativity" that is a part of model building and manipulation results in a detailed sensorium in which the modeler "feels" their way toward the correct structure.[38] In this light, fixating on "visual problem solving" as the human players' major contribution to the FoldIt ensemble appears uninformative or, worse, a gross oversimplification.

FoldIt's emphasis on visualization and collaboration recalls a number of features from Palantir Gotham and the human trafficking investigation discussed above. The software presents a procedural possibility space where the protein is rendered as a graphical object open to human manipulation according to a selection of "moves," the results of which are calculated and re-presented by the FoldIt environment. Both FoldIt and Palantir allow users to rapidly visualize the consequences of their hypotheses on screen – for instance, after each move, FoldIt highlights whatever new energetically frustrated areas may have arisen with the new configuration. The player or analyst becomes part of a circuit of feedback and revision with respect to the problem in question. Moreover, interpersonal communication, including chat windows within the software environments themselves and channels within the wider organization, overlays a space for collaboration and strategy exploration and development. The case of ZackS might offer a similar portrait, with the computerized chess engines presenting candidate moves that fit certain criteria, deliberation on them among the two human players, followed by rapid computation, evaluation, and visualization of the results of subsequent moves. We can see in somewhat starker relief some of the investigative rhythms of composite problem-solving systems like FoldIt, Palantir Gotham, and ZackS, with cycles of rapid, coarse-grained mechanized search, human-driven refinement of potential solutions based on digital representations, followed by mechanized evaluation of those human-provided refinements in the service of a final decision.

---

[34] Ibid.
[35] Ibid., 759.
[36] Natasha Myers, *Rendering Life Molecular: Models, Modelers, and Excitable Matter* (Durham, NC: Duke University Press, 2015), 74.
[37] Ibid., 102.
[38] Ibid.

Cycles of visualization, manipulation, and experimentation – these seem to be key aspects of FoldIt's effectiveness as a digital artifact and, recalling the ZackS assemblage, we can see a few overlapping themes. Kasparov writes of the humans of ZackS and "their skill at manipulating and 'coaching' their computers to look very deeply into positions effectively counteracted the superior chess understanding of their grandmaster opponents and the greater computational power of other participants."[39] But if we can imagine stepping into ZackS's room, what would we see? Two humans, three screens, some keyboards and mice, some number of windows for various applications open on those screens. The two humans consider the reasonable moves for the current state of the game, input those into the chess engines, look at the results, compare, discuss, perhaps repeat, and select what they consider to be the strongest move. This space of interaction parallels what we see in the FoldIt example in that it creates an opportunity to ask, persistently, "well, what if we did this?" and have the computer (or computers) rapidly spin out and visualize the (rule-governed) consequences of that action. And, in comparison to the computer-aided grandmaster, the ZackS system prominently features difference and parallelism – four different digital systems at the same time, using different logics, to suggest and evaluate potential moves and two humans working with the computers, and each other, to choose the best solution. We might think of this as a chaotic, disordered situation – again, one Palantir speaker imagines with some surprise the "crazy ballet" presumably required[40] – but it seems like the heterogeneity of ZackS is actually a major strength. The heterogeneity of these systems – the creation of external models, visual or verbal, and the – communication between human and computer or between humans – creates a space of interruption and the possibility of noticing new aspects of the problem and alternative solutions. The success of these arrangements does not, as Licklider would have it, primarily derive from a thorough symbiotic integration of humans and computers where "clerical" systems seamlessly serve "intellectual" operators. Rather, the effectiveness of these systems, their creative potential, seems to depend on the space of play and disorder they create among subsystems that work together while communicating across a gap, an interface. A key component of this play, suggested by the case of FoldIt and Myer's account of protein modeling, may be the affective investment and bodily "reasoning" deployed by the human participants. These considerations raise new questions about designing problem-solving systems to court what Douglas Engelbart called "disorderly processes."[41]

4. Symbiosis (2): Problem Structure

Researchers' framing of FoldIt as a belated response to observed inadequacies of the standalone Rosetta algorithm raises a question: are such systems merely temporary, ad hoc arrangements for problems that are not yet understood well enough for fully automatic methods? Are these examples part of a minor, receding class of use cases? Are composite systems just a temporary tool because we don't know enough to fully structure certain problems for more completely automatic techniques?

---

[39] Kasparov, "The Chess Master and the Computer."
[40] Palantir Technologies, "Human-Computer Symbiosis" (video)
[41] Douglas Engelbart, *Augmenting Human Intellect: A Conceptual Framework*, (Menlo Park: Stanford Research Institute, 1962), Section II.C.5.e, "Flexibility in the Executive Role," http://www.dougengelbart.org/pubs/augment-3906.html.

The tractability of problems for algorithmic problem solving is often thought to be a function of their being either well- or ill-structured. According to Herbert Simon's 1973 article, "The Structure of Ill Structured Problems," well-structured problems (WSPs) are perceived having at least some or all of six features.[42] Based on these criteria, one can roughly evaluate the degree of structure for a problem as a function of the existence of definite (e.g. quantifiable) parameters together with the practicability of automatic or mechanized processes for computing the interactions among them. The case of protein folding, at first blush, would appear to fit most of these criteria for a well-structured problem. The free energy of a given conformation is a definite, quantifiable criterion for testing; bond angles between the constitutive residues form a problem space where all possible conformations can be represented, as well as transitions between those conformations. With predicting protein structure, it seems that the major difficulty arises with respect to the practicability of computation across an extremely large problem space. From this perspective, symbiotic human-computer systems should be mere interludes in evolution of better algorithms and more powerful hardware.

The remainder of Simon's piece, however, presents an interesting figure-ground reversal of this understanding. He claims that the success of artificial systems with respect to the idealized WSPs should *not* serve as a starting point for a general understanding of problem-solving and the design and evaluation of problem-solving systems. In Simon's view, the effort to impose the above features (and consequent frustrations) on all problems is misguided, a mistake that arises "when we systematically confound the idealized problem that is presented to an idealized (and unlimitedly powerful) problem solver with the actual problem that is to be attacked by a problem solver with limited (even if large) computational capacities."[43] Rather than ill structured problems (ISPs) forming a residual (and receding) class of problems, Simon suggests we see them as the primary class of problems, with WSPs forming a set of exceptional, idealized cases. In his words, "It is not exaggerating much to say that there are no WSPs, only ISPs that have been formalized for problem solvers."[44]

Departing radically from the apparently well-structured examples of chess and theorem proving, Simon turns to architecture – the design of a house – as an instructive example of problem-solving that occurs despite the supposed "ill structured" nature of the problem space. Pointing to the fact that "problem solvers of familiar kinds can go to work even on problems that are, in important respects, ill structured," Simon suggests that "perhaps we have exaggerated the essentiality of definite structure for the applicability and efficacy" of mechanical techniques.[45] His description of the ill-structured house design problem highlights the cyclical nature of this process, the importance of external representations of the current state of the problem space, and the key role of a noticing/evoking/retrieval mechanism in restructuring the problem space (and its representation) and prompting the call up of relevant information from long term memory. This process alternates "between problem solving in a (locally) well structured problem space and modification of the problem space through retrieval of new information from long-term memory."[46] The noticing and evoking mechanism has the ability to recognize salient features of the current representation or model of the problem and *interrupt* the problem-solving process with new information – criteria, subgoals, design generators – stored in long-term memory. More

[42] Herbert Simon, "The Structure of Ill Structured Problems," *Artificial Intelligence* 4 (1973): 183.
[43] Ibid., 186.
[44] Ibid.
[45] Ibid., 187.
[46] Ibid., 192.

concretely, the working architect creates plans and models, and in "viewing a model, the architect can detect relations among components of the design that were available to him directly from his plans."[47] Such models are not predictive in any exhaustive way, but the shift between different representations of the problem space is essential for the progress of the design process. The architect works within a system of heterogeneous elements, and the critical feature of these systems is that "the problem representation can be revised continually to take account of the information—of the real situation—so that the problem solver is faced at each moment with a well structured problem, but one that changes from moment to moment."[48]

With such a system in view, Simon revises the classic understanding of chess as a well-structured problem. Chess appears well-structured in the context of a single move, but the entire game involves a continual redefinition of what the problem is and the recognition of different salient features of the problem space. Simon puts forward a design for a chess program that takes seriously the construal of chess as an ISP, a design with three principal components: a move generator or "recognition net" that evokes moves to be considered in a given position; an evaluator that predicts and compares the consequences of the considered moves (and makes the "best" move); and an updating process that records that move and the opponent's reply before handing control back to the move generator/recognition net.[49]

Our phenomenological observations of computer chess programs may hinder our understanding of the problem-solving processes occurring therein; for instance, a computer's engagement with the problem space does not pass through "external" models or representations like the architect's. It seems that to the degree a problem-solving system remains opaque, working with more or less internal representations of the problem, a homogeneous calculative or deductive process is presumed. But what I take from Simon's inquiry into problem structure is that problem solving is not a predominantly calculative activity. Indeed, the calculative processes characteristic of WSPs, in Simon's view, can only follow on the construction of an idealized version of the problem. Priming the noticing and evoking mechanism is just as (if not more) important than the system's calculative capabilities. Moreover, this implies that intermediate representations must have some quality that drives transitions between problem subspaces and other intermediate representations. On the whole, this is a provocative suggestion and a consequential one for the study of computing machinery and knowledge practices. To speak of "finding" a solution to a problem may only be appropriate for a small set of idealized cases. Rather, we should say that solutions are primarily the result of *design*—and design entails aesthetics, alternatives, and decisions or, put otherwise, culture and politics.

5. At the interface: Graphs, Networks, and Social Knowledge

Simon argues that evocative representations or models of a problem space are crucial to the problem-solving process, a cyclical endeavor that is more akin to design than it is to calculation. In the case of Palantir Gotham, this "external" memory is dominated by the Graph application, and this visual device is not incidental – it is a structuring element of the procedural interaction between analyst and platform. Palantir claims that its platform makes it easier to ask the

[47] Ibid., 188.
[48] Ibid., 195.
[49] Ibid., 196.

questions you want, and, to a great degree, these queries are organized by the graph visualization. The human user looks at a subset of data visualized as a graph – nodes and edges, animations indicating flows, and icons identifying discrete but linked entities. New objects appear on the graph of a result of analyst prompts to "link by" or "search around," querying the system as to what other records have attributes linked to a given object on the Graph. The commensurability among different objects displayed on the graph (cell phones, bank accounts, persons, addresses) recalls Bruno Latour's concept of immutable mobiles and, in particular, optical consistency as a requirement for modern knowledge practices.[50] The Graph application is a space where an analyst can "muster" different digital traces as visible objects with relationships to convince – first themselves, then others – of an investigative hypothesis. It is a way of "talking *with* many absent things presented all at once."[51]

Latour's materialist account of early modern visual culture emphasizes the emergence of a common meeting ground, on paper, created by the techniques of linear perspective and the Dutch distance point. Such a meeting ground was key to the advance of scientific reasoning and the expansion of European colonial power in the early modern period. Optical consistency does not inhere in any objects or data but is produced by certain inscribing procedures. These techniques permitted the regular and convincing combination of images from many sources, both natural and fictional. (In the case of Palantir, we might say known and hypothetical.) Palantir, both in mythos and method, attempts a similar feat in the way it accumulates space and time, resulting in a synoptic presentation of a subset of data "objects" for the analyst. Taking Foucault's argument in *The Birth of the Clinic*, Latour reiterates the importance of such presentation and claims that "the same medical mind will generate totally different knowledge if applied to the bellies, fevers, throats and skins of a few successive patients, or if applied to the well-kept records of hundreds of written bellies, fevers, throats and skins, all coded in the same way and all synoptically present."[52] Latour argues that medicine, among other disciplines, becomes scientific not in the mind but on paper. Knowing "more" means seeing more things, many things, *at once* – but according to a specified logic that provides a consistent surface for viewing.

The production of graphs entails the algorithmic reduction of a large matrix of relational data into a two-dimensional image of edges and vertices (lines and points) in order to make the data communicate certain information about higher level characteristics. With the development of computer displays and drawing algorithms, graph visualizations (also called network visualizations or sociograms) have become an increasingly popular form of representation for the growing quantities of relational data generated in and by sociotechnical systems. Though empirical studies of human cognition and information visualization have only started to gather momentum over the past decade, the history of social network research supplies a number of important "intuitive" starting points as to the representational and interpretive features of graphs. J.L. Moreno, an early figure in social network analysis, suggests that "as the pattern of the social universe is not visible to us, it is made visible through charting."[53] Klovdahl's survey repeatedly

[50] Bruno Latour, "Visualization and Cognition: Drawing Things Together," in *Knowledge and Society: Studies in the Sociology of Culture Past and Present*, ed. H. Kuklick (Jai Press, 1986): 1-40.
[51] Ibid., 8.
[52] Ibid., 15.
[53] Jacob Moreno, *Who Shall Survive? Foundations of Sociometry, Group Psycho-therapy, and Sociodrama* (Beacon, NY: Beacon House, 1953): 96, quoted in Alden Klovdahl, "A Note on Images of Networks," *Social Networks* 3 (1981): 198.

suggests that sociograms make the certain kinds of connectedness among actors "obvious" or "immediately evident," allowing the analyst to understand the influence, status, and other features in a way that the mere counting of contacts or relationships would not.[54] Moreover, manipulating aspects of the visualization facilitates the perception of different features from the same data. One major concern in graph visualization is the placement of nodes to enhance the visual clarity of the image. Specific principles for the drawing of social graphs include prominence indices, linked to concepts like status or influence, and cohesion principles that draw substructures (i.e. groups) in clusters based on levels of interconnection.[55] On the whole, social network analysts point to the ability of graphs to emphasize *structural* features and assist in the development (and communication) of structural insights from relational datasets.[56] These images resonate with concepts of material structure to evoke hybrid social-architectural notions like density or sparseness, centrality and betweenness, height and status, support and dependence. According to Brandes et al., such visualizations (and these attending notions) can help analysts answer questions like: "Who has the power?" and "What are the consequences of the power structure?"[57]

It is almost impossible to disentangle graph visualizations from their prominent role in the development of social network analysis – the network as a concept inherits much of its explanatory power from the visual metaphor of the graph, and vice versa. In particular, the commensurability enacted by graphs/network analysis hinges on a reduction of social reality to dyadic units, that is, actors and relationships. A prominent application of social network analysis comes from economic sociology, with Mark Granovetter's 1973 article "The Strength of Weak Ties" widely cited as an influential study in asserting the efficacy of this approach.[58] Linking norms to network density, cliques to openness to new information, and the strategic advantage associated with "structural holes," Granovetter and other network analysts see the possibility of understanding (and even predicting) macro-level social phenomena using simple data sets that count the relationships between individual actors.[59] Bourdieu's critique of these understandings of economic action highlights a few a salient peculiarities and elisions of network analysis and its visual exponents, which act as a mutually reinforcing pair. What Bourdieu calls the "interactionist vision" limits itself to direct, conscious actions by and between agents, without accounting for the field in which these interactions take place.[60] The field structures "the spaces of the possible open to" these agents, in the form of rules, regularities, and the differential strengths of the entities within it.[61] (There may be visual devices for representing such characteristics.) According to Bourdieu, the interactionist vision (metaphor of points and lines) ignores history and cannot capture the agonistic nature of economic action in a field and the effects of hegemony on the agents therein. Examining the strength of weak ties, in other words,

[54] Klovdahl, "Images of Networks," 199.
[55] Ulrik Brandes, Linton C. Freeman, and Dorothea Wagner, "Social Networks," Ch. 26 in *Handbook of graph drawing and visualization*, ed. Roberto Tamassia (CRC Press, 2013).
[56] Linton Freeman, "Visualizing Social Networks," *Journal of Social Structure* 1, no. 1 (2000): 4. Retrieved from http://www.cmu.edu/joss/content/articles/volume1/Freeman.html.
[57] Ulrik Brandes, Patrick Kenis, and Jörg Raab, "Explanation Through Network Visualization," *Methodology* 2, no.1 (2006): 18.
[58] Mark Granovetter, "The Strength of Weak Ties,"
[59] Mark Granovetter, "The Impact of Social Structure on Economic Outcomes," *The Journal of Economic Perspectives* 19 (2005): 33-50.
[60] Pierre Bourdieu, *The Social Structures of the Economy* (Cambridge, UK: Polity Press, 2005): 195.
[61] Ibid.

may not permit the analyst to properly see the actor who dominates not because of conscious influence but because "the structure acts on its behalf."[62] For Bourdieu, the concept of habitus provides something that network analysis cannot, namely an understanding of how action (including economic action) is: "an 'intelligent' response to an actively selected aspect of the real: linked to a history fraught with a probable future, it is the inertia, the trace of their past trajectory, which agents set against the immediate forces of the field, that means that their strategies cannot be deduced directly from either the immediate position or the immediate situation."[63]

These conceptual emphases and limitations are widely acknowledged by contemporary social network researchers, but, nonetheless, their explication serves as a useful starting for understanding graphs as a "design tool" for constructing explanations for observed events and behaviors. For the architect in Simon's account, part of the design/problem-solving process is the scrutiny of models of the projected structure, and the relations available for detection are a function of the characteristics of the visual model.[64] Analogously, the graph as a visual metaphor frames a set of possible questions and answers available to an analyst. The network images imply a fundamental reality of direct exchange among distinct social agents, eliding the concrete channels and contexts in which these actors are embedded and another instance of the construction of information as a bodiless fluid.[65] The patterns of these circulations in a seemingly thinned out present, understood as a network, are put forward as the defining aspect of social action. According to critics, this understanding may de-emphasize the embeddedness of social life, with little acknowledgment of history, context, or milieu. Domination and power, for instance, are understood as influence. What activates social network theory and research into graph drawing is the assumption and aspiration that these high-level visual devices will, if rendered properly, analogously indicate social structures of interest irrespective of the precise kind of relational data, social activity, and cultural context. Social network analysis and graph visualization researchers have long had exploratory ambitions for these techniques and have professed their potential to reveal "previously hidden facets of society." [66] The metaphorical gravity here is immense – the image "is" the network, the social structure. Observable, macro-scale phenomena are explained using the features of the underlying social network instance/image. In the world of the graph, normally unseen but now algorithmically drawn social networks are thought to identify, explain, and predict the character and behavior of social groups and actors.

This brand of social epistemics entails a number of concerning features, especially in consideration of the institutional contexts in which such practices may be deployed. As mentioned above, the neglect of "field effects" may overstate the independence and power of smaller actors and understate that of larger actors and institutions. Single persons within a network may be presented as a discrete nexus of risk or danger (and a justified target of intervention) without consideration of how their responses are conditioned by the field and the dominant actors within it. Conversely, the dominance of larger actors is understated or even absent – in the Palantir investigations, the "small" enemy is much more prominently visualized than the larger organizations of the United States defense apparatus – opportunities for

---

[62] Ibid.
[63] Ibid., 212.
[64] Simon, "Ill Structured Problems," 188.
[65] N. Katherine Hayles, *How We Became Posthuman* (Chicago: UChicago Press, 1999): 50.
[66] Moreno, *Who Will Survive?*, 96, quoted in Klovdahl, "Images of Networks," 198.

reflexivity are minimized. An inherent antagonism with privacy emerges here as well – better understanding of actors and social structures hinges on knowledge of second- and third-order relationships (at least) and the inclusion of data about actors that may be quite distant from the behavior or actors in question. With social network analysis, the best way to know any one thing is to know everything, to have the total graph available, at least potentially, for viewing and manipulation to discover latent patterns. The crucial object of knowledge regards what practitioners refer to as "community structure," a knowledge which is valuable both in itself and for situating the behavior of individuals of concern.

6. Combating Dispersion: Networks and Narratives

At the beginning of Palantir's human trafficking presentation, the engineer makes two linked claims about the graph object model. After importing bank transaction records from a spreadsheet and displaying them on the graph, she claims that "we model data the way human beings actually think about the world." Shortly thereafter, she claims that is not only more compatible with human cognition but is indeed "the way the world really works."[67] The network as analytical concept promises the ability to unveil hidden webs of influence and action-at-a-distance in social life. Alongside, the network as social form suggests that the social fabric is shot through with invisible structures of power and influence, an arena of spectral para-politics filled with figures like rogue states, terror groups, and hacktivist organizations. Networks emerge as an urgent problem of knowledge for institutions as the concept moves from an epistemological tool to an ontological class and becomes reified as a fundamental social form characterized by unique kinds of opacity and hazard. Palantir's Gotham environment and surrounding discourse reflect and perform this supposition of a world of networks, asserting that significant organizational knowledge must be framed in terms of networks and enjoining organizations to, as a matter of survival, investigate and understand these risk-laden social forms. The specifics of this knowledge problem (and its solutions) are not, however, always clearly articulated. What does it mean for the kinds of institutions in which Palantir is deployed to pursue knowledge of "community structure" in and through the image-concept of the network?

Military and defense thinkers have been particularly challenged by the emergence of networks and network-centric warfare, according to Samuel Weber.[68] Engaging primarily with a 2001 RAND publication by Arquilla and Ronfeldt, *Networks and Netwars*,[69] as an exemplar of this discourse, Weber traces efforts to create actionable definitions of networks in defense thinking. Arquilla and Ronfeldt identify the novel effects of networks that confound traditional military strategy: swarming, for instance, "occurs when the dispersed units of a network… converge on a target from multiples directions… then dissever and redisperse, immediately ready to recombine for a new pulse."[70] Networks aim at "sustainable pulsing" in their application of force, a principle that differentiates them from traditional military and guerrilla strategies of concentration. Such effects emerge from the counterintuitive nature of networks as the source of

[67] Palantir Technologies, "Building a Human Trafficking Case" (video).
[68] Samuel Weber, "Targets of Opportunity: Networks, Netwar, and Narratives," *Grey Room* 15 (2004): 6-27.
[69] John Arquilla and David F. Ronfeldt, *Networks and Netwars: The Future of Terror, Crime, and Militancy* (Santa Monica: RAND, 2001).
[70] Arquilla and Ronfeldt, *Networks and Netwars*, 12, quoted in Weber, "Targets of Opportunity," 14.

dispersed yet coordinated action and present a new intelligence problem. Networks are elusive because they operate with some hitherto unknown principle of cohesion, one that seems to wink in and out of visibility.

For defense planners, the knowledge problem posed by networks stems from their lack of conventional coordinating structures – clearly articulated chains of command, supply lines, visual identifications – that military intelligence and strategy have historically relied upon. In particular, these characteristics frustrate *targeting* as a basic element of military action – the positive identification of an enemy.[71] In place of these structures, Arquilla and Ronfeldt identify a novel principle of coordination in network: the circulation of narratives, or what they call doctrinal leadership. These stories "expressing the netwar, and the doctrine guiding its strategy and tactics" are the protocol that links the dispersed members of a net, communicating the *who*, the *how*, and the *why* of a group and allowing it to function without clear hierarchies, command, and discipline.[72] In this reading, a given network is an effect of some shared narrative in circulation that brings together characters (us and them) and criteria for identification with motivations and principles of action. The network as knowledge problem becomes tractable when strategists realize that these structures are held together by common notions of origin, style, motive, and action – when they are seen as a rhetorical effect of storytelling.

The military analyst engaged in understanding a (purported) enemy network now has a concrete task: to uncover the specific narrative that defines that network and forms its principle of cohesion and to situate correlated phenomena, observed activities and outcomes, as effects of these circulations. The narrative – or, rather, the fact of the narrative and its being shared – marks a network, which in turn explains the observations. Reading off the belated traces of the network from phone call metadata, bank transfers, common addresses, and the like, the analyst reconstructs a narrative whose elements serve as an explanation for these and other events. Specifically, a social network analysis or graph investigation (e.g. Palantir Gotham) presents these elements as actors and relationships, a presentation whose effectiveness assumes that a pattern of connection can be reliably linked to the existence of a coordinating principle. Viewing actions as a function of position in a network structure allows an analyst to propose the rudiments of a story, the barest of plotlines, though at the cost of historical depth and other contextual information, which is either lost or must be added later as a supplement. The reconstruction of this narrative draws from a pre-existing catalog defined by the training and institutional position of the analyst and is a necessarily motivated rendering of any potential "real" narrative or coordinating principle.

Despite the potential limitations of this approach to understanding network-centric warfare, the coordinating role of doctrine and shared narrative provides a framework with which to understand the aspirations of social network analysis. Defense thinking around networks articulates the specific ignorance generated by network organizations, how that ignorance can be mitigated, and a sense of the institutional stakes of this modern social form. Arquilla and Ronfeldt sketch an explanation for how these structures emerge and, moreover, the beginnings of a rubric for their institutional evaluation. What distinguishes one network from another (or an instance of some other organizational form) is the *narrative* that guides its dispersed activity. To recognize a network is to properly identify the narrative that informs and makes sense of what, at first glance, seem to be unrelated actions. In the age of network-centric warfare, intelligence and

[71] Weber, "Targets of Opportunity," 14.
[72] Ibid., 15-16.

investigation establish narrative – origin, motive, style, and capability – for a dispersed group and re-enable the choice of determinate targets. From the analyst's perspective, such narratives primarily concern the collective network structure and its roles—the persons involved do not appear, as they do in traditional narrative, as characters whose choices advance the plot.[73] In a sense, people appear as events external to the network and its narrative that extend and modify the network through their participation.

Put otherwise, the contemporary challenge for institutions is undoing these networked convolutions of identity, convolutions borne in part of modern information and communication technologies, that frustrate conventional systems of intelligence gathering and security. The development of Palantir suggests that at the current technological moment computing machinery, working in relative isolation, cannot effectively associate data points based on mechanism and narrative, and hence is inadequate to the challenges posed by networked threats. Working with a human analyst, however, the machine in the Palantir environment facilitates the rapid visualization of a number of possible data subsets – possible graphs – for the human components of the system to work on using their own sense of mechanism and story to unravel the obscurity of such networks and reveal their elusive organization.

---

[73] Seymour Chatman, *Story and Discourse: Narrative Structure in Fiction and Film* (Ithaca: Cornell University Press, 1979), 53.

7. Abduction: Diagnosing the Network

In their focus on interaction and relationship, network concepts and graph visualizations valorize questions about individual position (choke points, influencers, hubs) and about collective structure (the existence, identity, and closeness of groups and cliques). But graphs need not always be static representations – they are also tools of discovery. The "exploratory" potential of the graph is highlighted by the cycles of addition, subtraction, and rearrangement performed by analysts within the Palantir Gotham environment in an effort to perceive some elusive network structure.

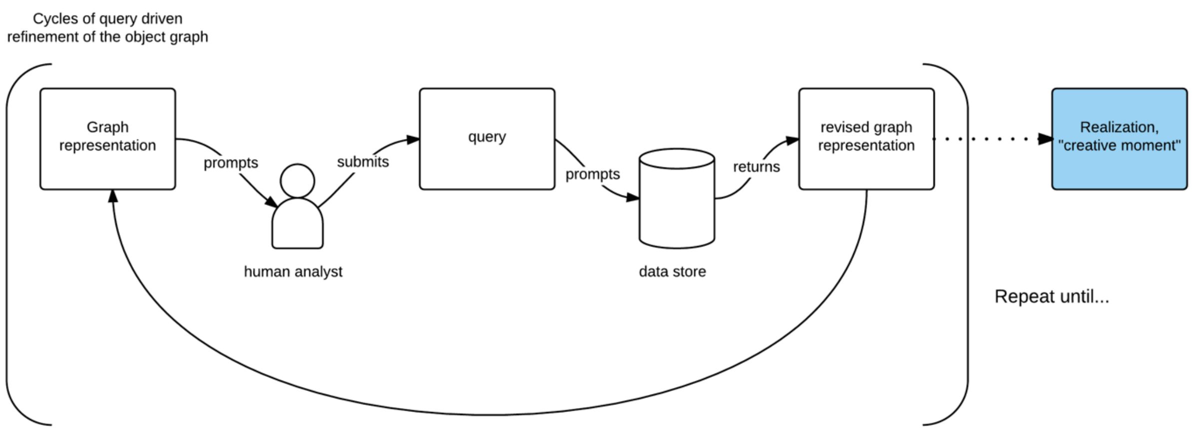

Figure 1.7. Diagram of analytical cycle in Palantir investigation. Author illustration.

This cyclical, visual exploration of subsets of data to design solutions recalls Simon's representation of the problem-solving process[74] and can be described as an instance of a more general (but also contentious) category of cognitive processes logician C.S. Peirce termed *abduction*. Distinct in logical form from induction and deduction, "abduction is the process of inferring certain facts and/or laws and hypotheses that render some sentences plausible."[75] Often summarized as "inference to the best explanation,"[76] abduction is said to encompass a range of epistemic performances (including scientific discovery and medical diagnosis) where cognitive agent relates observed phenomena to some unobserved causal structure. It is the act of making sense of a co-related set of observations. Medical diagnosis, in particular, presents a familiar example of such cycles of observation, hypothesis, testing, followed by new observations, revised hypotheses, and new tests. Abduction brings about presumptive rather than actual solutions to problems, with these hypotheses judged in terms of features like range, coherence, or consistency (rather than truth *per se*).[77] Many sets of observations entail a "combinatorial

[74] Simon, "Ill Structured Problems," 192.
[75] Lorenzi Magnani, *Abductive Cognition: The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning* (Berlin: Springer, 2009): 8.
[76] J.K. Fetzer, Artificial Intelligence: Its Scope and Limits (Dordrecht: Kluwer Academic, 1990): 103, quoted in Magnani, *Abductive Cognition*, *7*.
[77] Magnani, *Abductive Cognition*, 11.

explosion of alternatives" in terms of explanation, and abduction can be seen as an economical (rather than logical) response to an "ignorance problem."[78]

Cognitive scientist Lorenzo Magnani introduces a useful distinction between off-line or theoretical abduction (which occurs more or less "inside the head") and online or manipulative abduction. Manipulative abduction makes use of external objects ("epistemic mediators") like models, pictures, and diagrams to inspire a kind of "discovering through doing."[79] Akin to Simon's description of the architect's model as a kind "external memory" serving as the substrate for the designer's noticing/evoking mechanism, Magnani contends that external objects and visual devices "are not merely memory aids: they can give people access to knowledge and skills that are unavailable to internal representations."[80] Specifically, external representations enable an alternative range of aspectual perceptions or construals of the data and the problem space – they "constitute a provisional creative organization of experience."[81] In manipulative abduction, hypothesis generation and testing are entangled with the construction of construals, hypothetical interpretations of experience, and this process of building relies on "the strategic application of old and new templates of behavior connected with extra-theoretical components, for instance emotional, esthetical, ethical, and economic."[82] The construction and manipulation of external epistemic mediators elicit interesting features from the problem space, but these activities do not passively convey a "logical" portrait of the information at hand. Manipulative abduction proceeds according to disciplinary and cultural templates of "epistemic doing." The process as a whole, according to Magnani, is economical and, moreover, highly conjectural.

But what about a given image prompts the analyst to ask another question or initiate a manipulation of the graph and the subset of data represented there? In the narrated demonstrations, one crucial element in this cycle is analogy (or disanalogy): the analyst's sense that "this looks like…" User hypotheses about the patterns behind the data prompt either: a search for records that would revise the graph and extend, confirm, or disconfirm the hypothesis; the removal extraneous objects from the graph or the rearrangement of objects on the graph. All of this with the goal of more clearly seeing the potential visual analogy, the obscured social structure that generated these digital traces. The graph "not looking like anything" may also prompt the addition of objects, visual re-arrangement, and other actions to explore and foreground (hopefully) latent patterns in that subset of data. These query cycles, the adding, pruning, and arrangement of the Graph, visually re-situate the data points and court analogical moments like "money doesn't usually move in such structured ways," spurring the hunches and hypotheses that carry the investigation closer to a more conclusive explanation of the data. Palantir, as an artifact, serves as a platform for analogical thinking about digital traces by human analysts in terms of the graph, which operates as an epistemic mediator both visually and conceptually.

Examining Palantir Gotham investigations through the lens of abductive cognition (in particular manipulative abduction) emphasizes the dynamic character of interactive data analysis using the graph visualization. The concept of abduction usefully highlights discrete phases and attendant features of the cognitive processes gleaned from the Gotham demonstrations, along with highlighting the end product – a plausible hypothesis or diagnosis, a reasonable explanation

---

[78] Ibid., 20.
[79] Ibid., 11-12.
[80] Ibid., 45.
[81] Ibid., 50.
[82] Ibid., 48.

– that makes (limited) sense of observed behaviors and abstracted social structures. The interactive graph visualization and its supporting data sets form the focal point for building construals in the Palantir Gotham environment. As an epistemic mediator, the graph at turns: elicits conjectures (according to both domain-specific and, presumably, larger cultural templates); provides an environment for their construction; and sets the stage for their embodied visual-conceptual evaluation. In the language of medical reasoning, the analyst in Palantir manipulates the graph in an effort to produce the best diagnosis. In the human trafficking demonstration, this diagnosis brings together observed outcomes (the initial set of suspicious bank transfers) and an abstracted network structure (the actors and relationships represented on the graph) with a plausible classification for the structure. As with medical diagnoses, these classifications (human trafficking network, cyber crime ring, terrorist cell, etc.) are drawn from an established catalog of entities, catalogs that reflect contingent and historically situated understandings of a domain. The manifestation may be more or less complex, or obscure, but the presumption is that the social phenomenon (outcomes/structure) in question is an instance of a known class of social organization.

As I outlined above, critiques of social network analysis and graph imagery commonly focus on the way in which the technique tends to flatten social life into independent actors and their interactions, eliding questions of domination, history, and other embedded constraints on behavior within a social field. It is tempting to characterize graph visualization as an instance of such "flat" approaches to social knowledge and another iteration of the inhuman digitization of knowledge and social life. Dismissing network visualization techniques and analysis based on their reductive nature, however, misses the ways in which these flat images support, enable, and limit knowledge practices. The network image-concept serves as an epistemic mediator, a productive element in the ampliative process of abduction. Similarly, a growing literature has emerged around providing specific accounts of how digital methods and imagery "*thicken* as well as *flatten* worlds."[83] Mackenzie and McNally, in their analysis of the use of cluster heat map visualizations in proteomics, argue that such flattened figures are sites of superimposition, recalling Latour's discussion of early modern linear perspective and the "common meeting ground" for variegated phenomena. What Mackenzie and McNally call the "thickness of digital devices" comes to the fore in Palantir's deployment of graph visualization as a device that preserves certain kinds of multiplicity to aid in the search for identities.[84] The Palantir graph may sometimes provide a synoptic view of social structure, but, for the most part, it is a site of mustering, layering, and revision – new relational datasets are added, links and nodes are refigured, temporal and geospatial information augments the analysis. This "interlacing of phenomenal levels," this thickening and flattening, occurs both in the abductive movements at machine-analyst interface and within the database systems and their ontologies which I examine in Chapter Two.[85]

In the end, many Gotham investigations strive for the identification and characterization of some actor or group of actors – membership, roles, motives, and capabilities – in order to comprehend the threat they may pose to an organization. Palantir commonly employs the trope of "situational awareness" to describe the value-added of their data platforms, promising an

[83] Adrian Mackenzie and Ruth McNally, "Living Multiples: How Large-scale Scientific Data-mining Pursues Identity and Difference," *Theory, Culture & Society* 30, no. 4 (2013): 73.
[84] Ibid., 87.
[85] Ibid., 76.

understanding not just of any single threat but of the landscape of potential risks and vulnerabilities. Lucy Suchman's analysis of the discourse of "situational awareness" in drone warfare describes the underlying knowledge problem in this domain as the "(mis)identification of relevant others" on the battlefield.[86] Suchman's account traces the broad assemblage of social and technical systems involved in UAV (unmanned aerial vehicle) missions in the Afghan War, including camera operators, pilots, commanders, observing ground forces, and intelligence specialists – an ensemble of dispersed agents working to classify an anonymous body, seen through a lens thousands of feet in the air, as a civilian, friendly, insurgent, or "imminent threat." Though the precise circuits that bring information from analysis platforms like Palantir to drone control rooms are likely to remain classified, U.S. officials have left no doubt that network analyses play a major role in "the configuration of 'their' bodies as targets to be killed,"[87] that, in the words of former CIA director Michael Hayden, "we kill people based on metadata."[88] These assertions about these others, their communities, their community structures come via abductive cognition and are not logical deductions from self-evident social data. Abduction highlights this investigative work and the "situational awareness" it engenders as, indeed, an act of configuration, a "crazy ballet" that coordinates humans and machines, digital traces and visual representations, narratives, categories and conjectures in an epistemic performance that should be described, in no uncertain terms, as extra-theoretical.

8. Conclusion

For the Palantir platform, the Graph is both the central process and product for understanding "anomalous" sets of digital traces. And the Graph as process and product, embodied in these evolving representations, provokes a kind of gestalt recognition of previously absent or elusive organizational structures. Here, human users do much more than tend the computing machines engaged in an indifferent process of algorithmic knowledge production. Rather, the human has a unique and active role, one that Palantir engineers explicitly thematize in their discussion of Licklider and his work on human-computer symbiosis. In particular, in one of the talks on human-computer symbiosis and friction described above, a Palantir speaker highlights the protein puzzle-solving "game" FoldIt, drawing our attention to the "playable" nature of the Palantir investigative platform. We see knowledge practices that are articulated with a range of disordered and exploratory "moves" – clicking and dragging, shaking and wiggling, adding and deleting, arranging and rearranging. These moves feed build to the key phenomenological moment. At some point, a human analyst looking at the object graph has a moment of realization. The visual arrangement of entities reveals some instance of organization that was once elusive, and this realization, through narrative, situates and gives meaning to the original anomaly that provoked the investigation.

This creative moment comes about through "play" in a space that seems designed to court disorder – there are false starts, dead ends, additions and deletions, rearrangements. The possibility space here is not merely rhetorical, nor merely speculative – it is *effective*, creating,

---

[86] Lucy Suchman, "Situational Awareness: Deadly Bioconvergence at the Boundaries of Bodies and Machines," *MediaTropes* 5, no. 1 (2015): 2.
[87] Suchman, "Situational Awareness," 6.
[88] Hayden as quoted in David Cole, "'We Kill People Based on Metadata,'" NYRDaily Blog, May 10, 2014, *The New York Review of Books*, http://www.nybooks.com/daily/2014/05/10/we-kill-people-based-metadata/.

according to its authors, new insights. It participates in a performance of intelligence and knowledge production. The Gotham environment makes anonymous and pseudonymous accounts, loci of digital activity, into named entities based on their position in a graph image of other transacting accounts – whether those transactions are phone calls, bank transfers, or e-mails. What Palantir claims is to have brought the analysis of digital traces, and anomaly detection specifically, into a more intensely human possibility space, one that engages visio-spatial cognition, and that this space – this composite or symbiotic intelligence amongst human analyst and computing machinery – results in the faster extraction of higher quality knowledge from massive datasets. This is an evocative computing that depends on a deliberate disorderliness in the service of a greater "situational awareness" for organizations. In effect, the platform works to de-anonymize the users of these systems and assist analysts in generating narratives that explain the relationships among anomalous/anonymous digital traces and classify associated entities. With respect to the various security domains in which Palantir is deployed, this mode of power/knowledge gestures toward the expansion of an interventional logic of "precision strikes," as institutions increasingly aim to reduce exposure, expenditure, and "collateral damage." Our understanding of contemporary forms of governmentality and control must include more expansive understandings of algorithmic reason and respond to computing machinery's inflections (rather than determinations) of the interpretive work that constitutes objects and subjects of technical and political action.

Finally, I would like to take up an earlier gesture with respect to the work of J.C.R. Licklider. As I described above, the fundamental distinction in "Man-Computer Symbiosis" is between clerical activities ("thinking") – for instance, calculating and plotting graphs, that is, "getting in a position to think" – and intellectual activities. The clerical and the mechanical are concerned with "preparing the way for a decision or an insight" while intellect supplies the decision or insight itself.[89] Licklider's vision calls for the automation and acceleration of the former activities in the service of shortened intervals between the all-important moments of decision. Though Palantir emphasizes Licklider's symbiosis as a conceptual antecedent for their design thinking, the heterogeneous elements and the proliferation of interfaces in the case of ZackS suggests a different approach to the environments and workflows at work in these exemplars. Friction connotes a scalar factor, a *post hoc* measure of less as better and more as worse with respect to a valorized moment of human decision or insight. A more expansive view is seen in Douglas Engelbart's *Augmenting Human Intellect* and the understanding of synergy that he elaborates therein.[90] An analysis of these systems in terms of synergy highlights the ways that friction (as contact and movement) may be productive and turns our attention from evaluating the amount of intelligence in these systems and instead toward the different hierarchies of capabilities and the kinds of intelligence each display. Moving from symbiosis/friction to synergy better accommodates the examples (both the ZackS chess playing assemblage and the FoldIt protein folding "game") as instances where human-machine performances outstrips less cooperative systems, as these systems, in their use of multiple agents, visual imagery, and embodied manipulation and experimentation, suggest a design approach to the contact and movement of heterogeneous elements that goes beyond improving "match" and "fluidity." Seen through the lens of synergy, these human-computer assemblages demonstrate how these points of contact, sites for the transfer of energy and information across an interface,

---

[89] Licklider, "Man-Computer Symbiosis," 6.
[90] Engelbart, *Augmenting Human Intellect*, "The Source of Intelligence," Section II.C.1.

might amplify and transform the intellectual capabilities of human beings.[91] The examination of the Gotham environment here has focused on how the Graph as interface, as possibility space, provides opportunities for disorder and play, hypothesis generation and abductive inference, and is uniquely productive of the insights that follow. The composition of the Gotham environment suggests that these tools are not just algorithmic, but also interpretive, and deploy analogy and narrative sensibilities together with computation and calculation.

I would argue, then, that the intelligence of a composite system is a product of what happens at the interface, at that place where human and artifact navigate differences in materials and related differences in possibilities – the possibilities offered up by mammalian nervous systems and semiconductors articulated with one another via LCD screens, keyboards and mice, office environments, and organizational contexts. Intelligence emerges from the ways which, unpredictably, each reaches the other across a break. Understanding composite, human-computer systems becomes to a degree a materials science. When a system foregrounds the human component (which Licklider describes as a "noisy, narrow-band device" but one with "very many parallel and simultaneously active channels"[92]) in a computational process, opportunities for visio-spatial cognition, analogy, and narrative seem to find their way into that system's performance. The interface, the site (or sites) where matching processes take place, is what permits the information to change phases, to move from a silicon semiconductor medium to human nervous tissue and back again. Different types of reactions are energetically favorable in different media, and the consequent "synergetic structuring" physical processes,[93] the diverse activity of transistors and nerve cells, is central to understanding the character of resulting intellectual processes. Chapter Two will continue this analysis by examining the design of the database systems that support the interactive Palantir environments and developing a more liberal conceptualization of the interface by detailing the computational processes that organize and structure information prior to its visual representation. In other words, we must keep in mind that the procedural character of digital artifacts extends beyond *our* procedures for interacting and manipulating them. The form of these possibility spaces is defined by 'deeper' computational structures, and the contents that populate them are the products of data storage and processing techniques whose absence from the screen belies their infrastructural effect on these spectacular moments of knowledge, discovery, and insight.

[91] Ibid., "Concepts, Symbols, and a Hypothesis," Section II.C.4.
[92] Licklider, "Man-Computer Symbiosis," 6.
[93] Engelbart, *Augmenting Human Intellect*, "Capability Repertoire Hierarchy," Section II.C.5.

Chapter 2
From Digital Traces to Narrative Objects: Digitizing Ontologies for Data Analysis

1. Introduction

The contemporary rhetoric of intelligent organizations foregrounds a longstanding interpenetration of computing machinery and regimes of power-knowledge, however with a focus on the oracular potential of digital traces *en masse*. The previous chapter opened a discussion of Palantir, a data aggregation and analysis platform, as a recent exponent of this trend. In this composite system, we saw a computer that quickly retrieved subsets of records based on explicit criteria, coupled with a human analyst highlighting implied relationships to explore investigative hypotheses. The analyst, it seems, brings a human understanding of narrative and the mechanisms that may underlie the generation of some 'anomalous' subset of digital traces. Palantir's unique design rests on the hierarchy of processes which result in a machinic surface that: 1) provokes a *play* between computer environment and human analyst; 2) positions this play as a technique for problem-solving and the recognition of relevant entities; 3) facilitates the construction of narratives around these entities to move the investigation forward. The Palantir environment courts analogical and narrative thinking on the part of the human analyst; this choreography aims at the identification of elusive (and illicit) organization indicated by a subset of traces hidden within a massive dataset. However, the implications of a knowledge discovery and management system like Palantir derive not only from the spectacular moments of human-computer interaction, but also from the automatic, machinic configurations that collect, store, and retrieve these data on the "back end."

Game-like, visio-spatial experimentation and what Engelbart called "disorderly processes"[1] play a critical role in Palantir demonstrations and in illustrative examples they present: ZakS, the two-human, three-computer Freestyle Chess champion; and the Fold-It protein folding software. In describing their design approach, Palantir engineers cite J.C.R. Licklider's concept of *man-computer symbiosis* as guiding Palantir's efforts to create digital systems that amplify human problem-solving capabilities.[2] Licklider's sketch, however, seems to present the machine as only affecting the momentum of what is a fundamentally *human* intellectual process. The computer may accelerate or slow the intellectual activity of corporate presidents and military commanders, providing quantifiable benefits in terms of the speed and scope of decision-making, but the machine makes no qualitative difference.[3] The computing machine becomes a uniform surface defined by a single, *post hoc* measure, friction, and the machine is reduced to a static plane on which human intellection unfolds. Engelbart's theorization of the H/LAM-T (Human using Language, Artifacts, Methodology, in which the human is Trained) system and the synergistic nature of human-artifact interaction provides a more expansive model than Licklider, because it specifies the impact of artifacts on our physical capabilities for symbol manipulation. For Engelbart, composite capabilities (mediated at the man-artifact interface by matching process) are the true results of human-computer interaction.[4] The machine contributes more than a passive surface for the movements of human intellect. Rather, human-computer interactions

---

[1] Douglas Engelbart, *Augmenting Human Intellect: A Conceptual Framework*, (Menlo Park: Stanford Research Institute, 1962), Section II.C.5.e, "Flexibility in the Executive Role," http://www.dougengelbart.org/pubs/augment-3906.html.
[2] J.C.R. Licklider, "Man-Computer Symbiosis," *IRE Transactions on Human Factors in Electronics* 1 (1960).
[3] Licklider, 10.
[4] Engelbart, *Human Intellect*, Section II.A.

generate materially and formally unique intellectual processes and products. Engelbart's conception of synergy implies a qualitative transformation in the man-artifact assemblage that the concepts of symbiosis and friction do not.

Intelligence, as Engelbart describes it, is "elusively distributed throughout a hierarchy or function processes – a hierarchy whose foundation extends down into natural processes below the depth of our comprehension."[5] While attention to the surface, the man-artifact (or user) interface, is certainly one key consideration for Engelbart in the design and understanding of these composite systems (H-LAM/T), that interface and its matching processes are only a subset, the outward facing results of the mass of processes "within" the machine – its internal architecture. In describing the potential for digital computers to augment human intellect, Engelbart speaks of the multitude of symbolic "views" possible with computers and computer-controlled displays. Computers realize a multidimensional "internal image," and human users request certain aspects of that total image as desired.[6] A narrow emphasis on graphical software interfaces tempts us to describe our encounters with computing machinery as an abstract communion with disembodied information, eliding the materiality and politics of these devices and their histories.[7] Following Engelbart, we might reframe those encounters, or any user encounter, as limited, aspectual views into a total machine process. Accordingly, an inquiry into the politics of a digital artifact would extend beyond the user interface and the visual choices reflected there to the internal architecture of the system, an architecture that grounds the interactions and processes possible at the user interface. At various turns, Palantir emphasizes that their product is more than a visualization tool or environment. Rather, the Palantir platform is an "analytical infrastructure" with visualization as just the final layer atop systems for data integration, search and discovery, knowledge management, and collaboration.[8]

2. Abstraction and implementation: the uniqueness of digital possibilities

Engelbart's concern with the internal structure, dimension, and "convolutions" within digital artifacts finds a contemporary philosophical expression in theorist Wolfgang Ernst's analyses of media.[9] Ernst claims that the distinctive, microprocessual techniques at work in media form the "generative matrix" out of which macro-phenomena of history, culture, and politics emerge.[10] For Ernst, media archaeography investigates these processes as technological *archai*, the commandments or principles that materially govern the range of "what can be verbally, audiovisually, or alphanumerically expressed at all."[11] Novel *archai*, new microtechnological operativities, create new epistemological possibilities for a society, with Ernst comparing the phonograph cartridge and digital sound reproduction as one example of

[5] Engelbart, *Human Intellect*, Section II.C.1.
[6] Ibid., Section II.C.5.b.4).
[7] N. Katherine Hayles traces this struggle with the materiality of information in *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics* (UChicago, 1999).
[8] Palantir Technologies, "GovCon 7: Introduction to Palantir," YouTube video, 40:01, posted by "Palantir," Nov. 2, 2011, https://www.youtube.com/watch?v=f86VKjFSMJE.
[9] See Jussi Parrika, "Operative Media Archaeology: Wolfgang Ernst's Materialist Media Diagrammatics," *Theory, Culture & Society* 28, no. 5 (2011): 52-74.
[10] Wolfgang Ernst, "Media Archaeography," in *Media Archaeology: Approaches, Applications, and Implications,* ed. Erkki Huhtamo et al. (Berkeley: University of California Press, 2011), 252.
[11] Ibid., 239-240.

such a shift. The phonograph articulates the domains of mechanics, magnetism, and electricity, as the grooves of the record cause minute vibrations of the stylus, a magnetic pickup transforming them into a continuous electrical current for amplification and transmission. Contrast this with digital sound technologies, where sound waves are sliced and sampled, and the average amplitudes for a standard interval are encoded in sequences of electrical pulses, and stored and reproduced with the same logic. These two acoustic-electrical interfaces are characterized by different principles of energy transformation. In Ernst's view, this microprocessual changeover modifies our "cultural sonosphere" and creates new possibilities for storage, transmission, and transformation of sound.[12] Digitization profoundly modifies the "options of retrieval" for all manner of content, including sound, creating new possibilities for action and knowledge.[13] Among these possibilities are new techniques surrounding 'cultural' phenomena like music and computer graphics, but the digital *arché* also transforms the tools of governance in domains like surveillance, signals intelligence, CCTV monitoring, and facial recognition software. For Ernst, epistemology is really techno-epistemology, as a shift in the "hardware" of a culture works over the material bases for recording difference and similarity, for relation and reference, and for the perception and later manipulation of expressions. An artifact like Palantir implicates digital operativities with state and corporate power, and prompts us to examine how computational processes form part of the "reflexive prism" in which contemporary problems of governance and enterprise appear as problems of knowledge.[14]

Ernst's description of media archaeography as "a kind of epistemological reverse engineering" foregrounds an engagement with design, technicity, and machinic "configurations"[15] as a method for accessing the "subsemantic strata of culture."[16] Reverse engineering produces descriptions of a system after the fact, "reading off" the design of an artifact when aspects of its functionality are obscured. Reverse engineering may facilitate the duplication, improvement, or sabotage of a system, but its primary goal is a detailed understanding of the relations between means and ends – in short, the *design* of the system. Such an inquiry must engage with technical descriptions of artifacts, infrastructure and architecture; it must pass through the discursive surface and the "multimediatic interface illusion of the computer" to the design of the artifact.[17]

Design concerns the choice of parameters (ends) and command variables (means), and their organization to realize a possible, desired, but not currently existing environment.[18] The designer frames an end goal as a set of relevant parameters and values and chooses an optimal set of command variables that can, by means of certain material structures, bring the parameter values into the desired range. The design process results in a nexus of decisions to limit, as not all parameters may be relevant to the specified purpose of an artifact, and not all command variables are optimal or compatible with achieving the given parameter values. As a practice, design is characterized by a narrow focus on a limited task environment and the means to change limited aspects of that environment. With this framework, the first gesture of an "epistemological

[12] Ibid., 250.
[13] Wolfgang Ernst, *Digital Memory and the Archive* (Minneapolis: University of Minnesota Press, 2013), 126.
[14] Michel Foucault, *Security, Territory, Population*, ed. Michel Senellart, trans. Graham Burchell (New York: Palgrave Macmillian, 2007), 276.
[15] Ernst, "Media Archaeography," 239.
[16] Ibid., 242.
[17] Ernst, *Digital Memory and the Archive*, 125.
[18] See Herbert Simon, *The Sciences of the Artificial* (Cambridge, Mass.: MIT Press, 1996), especially chapter five, "The Science of Design: Creating the Artificial."

reverse engineering" would be to recover and recount this rational enframing, the delimitations of means and ends, that guide the construction of an artifact. A second gesture would explore how the artifact's microprocesses "overflow"[19] this rational imagining of a narrow task environment, and how those microprocesses form a matrix of possibility that may give rise to something more, or other, than the artifact's specified goal. The first gesture describes how the artifact is effective in accomplishing its defined object; the second suggests how those principles that govern an artifact's efficacy are generative of high-level sociotechnical phenomena, what Ernst might call epistemological *momenta*.[20]

Philip Agre defines the distinctive character and design potential of the computer as its ability, through the use of digital circuitry, to realize "any mathematical structure at all, as long as [the structure] is finite and enough money can be raised to purchase the necessary circuits."[21] This potential is entangled with the notion of *digital abstraction*. Digital abstraction distinguishes between the abstract definition of a computing artifact in terms of its digital logic (binary arithmetic) and the physical construction or implementation of that functionality. Each abstraction has a certain generality, in that it could be constructed using *any* physical elements that perform the necessary binary operations in the specified sequences and combinations. Much of the history of computing has been the extension of this logic, as engineers progressively package naked machinic possibilities into processor instruction sets, memory addresses, operating systems, applications, and files; a domain of abstract possibilities can be composed from the simple operations of digital circuits. As levels of abstraction proliferate, programmers have less need to consider "low level" particularities of a machine (such as the physics of the device circuitry) in creating new symbolic functionality. Over time, programming languages have developed high degrees of abstraction that allow designers to steadily "loosen the correspondence in time and space between the program and the machinery that implements it."[22] Theoretically, abstraction and implementation can be seen as reciprocally defined but independent levels of the computational process.[23]

In practice, however, a dialectical tension emerges between these two conceptual levels, and Agre suggests we might understand the history of computing as "the unfolding of dialectical interactions among various aspects of implementation and abstraction."[24] Using the figure of the wire, Agre outlines this dialectical tension in computing machinery. On the one hand, wire permits us to conceptualize arbitrary configurations of circuits, without necessary regard for the physical geometry and construction of the system. On the other hand, the finite speed of electrical transmission limits the efficiency of systems with extensive wiring networks.[25] In other words, it pays to formulate the digital abstraction with the possible physical implementation in mind and for designers to seek "a natural correspondence between the structure of a computation and its physical implementation."[26] And as computing systems introduce new, intermediate levels of abstraction, these constraints emerge at those higher levels of description as well. To that effect, Agre discusses the invention of indirect reference in computer programming,

---

[19] Michel Callon, "An essay on enframing and overflowing: economic externalities revisited by sociology," in *The Laws of the Markets*, ed. Michel Callon (Oxford: Blackwell, 1998): 244-269.
[20] Ernst, "Media Archaeography," 240.
[21] Philip E. Agre, *Computation and human experience* (Cambridge, UK: Cambridge University Press, 1997): 66.
[22] Ibid., 75.
[23] Ibid., 66.
[24] Ibid., 71.
[25] Ibid., 69-70.
[26] Ibid., 68.

whereby a memory address does not a store value but the address of another memory block. This indirection increases the flexibility of a system by permitting, for instance, repeated operations on different input values, rather than loading both the program and input values – putting the machine into a fixed configuration – each time it is operated. The cost, however, is an extra computational "step" to fetch the desired input value from the second address.[27] The ironic quip that "all problems in computer science can be solved by another level of indirection" speaks to this flexibility of digital computation, but practitioners know that all such indirection comes at some computational cost.[28] While "the computer engineer [...] can be assured of realizing any mathematical structure at all, as long as it is finite," the time and energy the computation requires will depend on the relationship the program has to the levels below and, ultimately, the physical circuity and its organization.[29] The design thinking proper to computing machinery begins with this tension between abstraction and implementation, which can be conceptualized as the drive to reduce the level of indirection in the system while still achieving a set of computational goals. Agre describes this thinking as an "architectural style of reasoning" that "attempts not merely to implement a given abstraction but to discover abstractions that both do the required work and admit of natural implementations."[30] With this understanding of digital computing, reverse engineering would then recreate the decisions embedded in the artifact – decisions about the expression of abstract, symbolic problems such that an efficient implementation can be constructed. In elaborating the internal architecture of digital artifacts, one describes the composition of the hierarchy of machine processes that form the substratum for the resulting symbolic action of the system.

Using the above notions of abstraction, implementation, and indirection, the remainder of this chapter details some of internal architecture of the Palantir platform and the presemantic, digital events that form the substratum for the hermeneutic work of a Palantir investigation. Specifically, the chapter will survey Palantir's design as a computerized datastore, a functionality that highlights a key convergence in the thinking of Ernst, Agre, and Engelbart. The capabilities of digital computers are inflections of their ability to realize various logico-mathematical structures, especially those mathematical spaces that index or map data resident in the computer's memory. This mapping and reference, as both Agre and Engelbart emphasize, is key to the functional efficacy of digital computers. A similar view informs Ernst's contention that the *mnemonic logic* of digital computers – the manner in which digital computers realize a space of memory addresses and "circulate discrete states"[31] – underlies the macro-historical effects of digital media, their impact on the semantic strata, and their peculiar techno-epistemological momentum.[32] The first step in the "reverse engineering" of the Palantir datastore consists in understanding the dialectic of abstraction and implementation embedded in the platform – particularly with respect to its abstract function as a computerized datastore, implemented using specific address schemes.

---

[27] Ibid., 72-73.
[28] Butler Sampson, quoted in Diomidis Spinellis, "Another Level of Indirection," in *Beautiful Code: Leading Programmers Explain How They Think*, eds. Andy Oram and Greg Wilson (Sebastopol, CA: O'Reilly, 2007): 279.
[29] Agre, *Computation and Human Experience*, 66
[30] Ibid., 83.
[31] Ernst, *Digital Memory and the Archive*, 138.
[32] Ibid., 139.

3. Palantir as datastore

A database (or datastore), generally speaking, defines both some abstract unit of related information, a record, and provides for the consistent storage and retrieval of these units.[33] A datastore's characteristics and practical efficacy are determined by the processes that group and translate abstract symbols into discrete microprocessual artifacts – how the system digitally implements the abstract notion of a record. The development of early database management systems hinged on questions of how, with a limited amount of memory and processor instructions, a system might, for instance, encode personnel records so that a user might find all the employees in a single department, or all employees with a certain supervisor, or all employees hired in a certain year. If this were presented in a familiar spreadsheet format, it would be a question of a method for identifying and presenting those rows whose values in the "Department" column matched the search criterion. In this case, we might say the abstract notion of an "employee" was implemented using a structure of rows and columns.

Consequently, one approach might involve a symbolic notation for row number, column number, and the associated contents to be stored in memory, and the implementation of such a structure using a digital computer. Maintaining datastores aligned with this tabular concept has been a focus of relational database development since the late 1960s. The contemporary mnemonic logic of computerized datastores centers on their processes for the generation and storage of identifiers (keys) and associated contents (values) – tuples, ordered sets of (presumably related) values – in memory such that, given only their identifier, the processor can later retrieve the contents. The values in these may also be encoded as references to other memory addresses, creating a structure akin to the index for an encyclopedia, which contains entry titles (identifiers or keys) and page numbers (addresses) that "point" the reader to the location of the full entry in a volume. (This indexing is critical because as data objects increase in size, the computational costs of retrieving each from memory and comparing it to the search criteria increases with them. Thumbing through an index is often far more efficient than looking through the book, sentence by sentence.)

From the 1990s onward, new developments in database management systems (including those at the core of Palantir data platform) increasingly looked to extend this mnemonic logic beyond the single processing element.[34] Innovations with respect distributed systems and cloud computing underlie the expansion of e-commerce, social media, video streaming, and many other applications on the modern web (Web 2.0). *Availability* becomes a key factor. With a single processing element fulfilling each request in order, an increase in the number of requests (because of more concurrent users, for instance) will result in the processor being less frequently available, as it executes instructions already underway. One approach to this challenge is *vertical scaling*. Vertical scaling refers to an increase in the power of the single processing elements – the more quickly the processor completes its tasks, the more likely it is to be available for the next one. However, the (very expensive) upper limit of vertical scaling is constrained by the current state of semiconductor technology – at any moment, only so many transistors can fit on any given processor, those transistors can only undergo so many state changes per second, and

[33] I use these terms interchangeably to refer to data storage systems in general. Due to the long dominance of relational paradigm, the term database has become difficult to separate from relational database. The term datastore has become a way to mark systems that may not use the relational paradigm.
[34] David DeWitt and John Gray, "Parallel Database Systems: The Future of High Performance Database Systems," *Communications of the ACM* 35, no. 6 (1992), 85-98.

energy consumption and heat dissipation continue to present practical challenges. Alternatively, distributed systems emphasize *horizontal scaling*, which increases the likelihood of there being an available processor by increasing the total number of processors rather than processor speed.

Horizontal scaling introduces the problem of what aspects of the database must be shared among individual systems in order to make them function, ultimately, as a single unit. Consider two simple examples of a distributed datastore. In one version, two (or more) independent machines have complete, identical copies of the dataset stored in memory. The coordinator keeps track of which machines are currently available to handle requests, and routes those requests accordingly. In this case, the coordinator requires no information about the location of any data, only the status (busy or available) of the various identical machines in the system. One drawback of this approach is that the system requires, in aggregate, storage that is many multiples of the size of the actual dataset – and this may be costly. A second approach would be to break up the data (rather than duplicate it) across the same number of machines. This division of the dataset increases the number of potentially available processors without making complete duplicates of the dataset for each machine. Fragmentation, however, introduces an additional layer of reference, as it requires that the coordination mechanism must have some "map" of which machines house the data associated with a given set of identifiers. In practice, most implementations simultaneously feature both duplication and fragmentation – a machine may serve as the primary node (or master) for a certain portion of the dataset, and a secondary node for another portion. What prompts this approach is the hypothesis that for the anticipated number of requests N, the average response time (the latency) t and/or the cost C of a system of a larger number of independent, coordinated machines will be preferable to a single, very fast system.[35] The mnemonic logic of a distributed datastore encodes relationships between identifiers and data objects (keys and values) not just on a single machine, but across a number of independent, networked machines. In this "shared nothing" paradigm, the coherence of the system is maintained only by the messages passed amongst the member nodes.[36]

Implementations of high performance relational database systems by companies such as Tandem, Teradata, and Oracle were a major focus of database development in the early 1990s. However, the proliferation of use cases that did not require the advanced operations (e.g. joins) or certain strict controls of relational systems – serving a web document differs greatly from the sorting and analysis of employee records – spurred further research into nonrelational paradigms as a way to increase performance. This "not only SQL" or "NoSQL" movement (with Sequential Query Language (SQL), a popular language for querying relational database systems, serving as a metonym for the relational paradigm), championed by a number of new players in the software and computing sector, were motivated by the possibility of simpler APIs, rapid tuning and loading of datasets, and more flexible approaches to database schemas.[37] Though there is some argument as to whether the promises of NoSQL have been fulfilled, by 2010 this approach to the problems of massive data storage has become an entrenched arena of development and research. As a more recent entrant into the data storage and analysis space, Palantir's platform is an inheritor not only of these specific technologies but also of their surrounding design concerns with respect to speed, extensibility, resilience, consistency.

---

[35] There is some dispute about which use cases this actually holds true for. See Andrew Pavlo et al., "A Comparison of Approaches to Large-Scale Data Analysis" (2009).
[36] Michael Stonebraker, "The Case for Shared Nothing," *IEEE Database Eng. Bull.* 9, no. 1 (1986): 4-9.
[37] C. Mohan, "History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla" in *Proceedings of the 16th International Conference on Extending Database Technology* (2013), 11-16.

A 2011 patent granted to Palantir entitled (appropriately) "System and Method for Investigating Large Amounts of Data" articulates the problem of database design with respect to the knowledge problems facing contemporary organizations, and it provides a comprehensive and concrete introduction to the abstract problems and chosen implementations at the core of their data platform.[38] "System and Method" highlights the risks organizations face from data exfiltration and fraud – problems of security – and presumes (as described in Chapter One) the choice of human-driven investigation to identify these anomalous, elusive, but highly damaging activities. Such analysis might include a user engaging in the investigation of "top-down trends, behaviors, and activities" or "bottom-up target centric analysis across a larger data set." The data that would ground such investigations, however, presents a number of challenges. Organizations' data sources are increasingly voluminous, dynamic, and heterogeneous, with possible sources including: "network traffic and access logs, bank transaction records, call data records, e-mail messages, netflows, electronic blogs, forums, [and] wikis." Palantir investigations involve iterative searches that filter the larger dataset so that a relevant subset can be visualized and explored without the "noise" of irrelevant data objects on the screen. An analyst only wants the 20 or 30 potential "needles" visible on the screen, without the surrounding bits of hay interfering with their visual thinking. More concretely, an analyst investigating a cyber attack may be looking for a "100 byte snippet of a single web access log" within a petabyte of web access data, or a fraud investigation may be hunting for "a single e-mail message amongst hundreds of thousands or even millions of messages." Palantir's investigative cycle of data retrieval and visualization, analogical thinking, and hypothesis generation calls for an architecture that can quickly find records matching an investigator's criteria from these various "haystacks" – an architecture centered on the fast and fine-grained search of massive, heterogeneous datasets. We can elaborate the mnemonic logic of Palantir as, in part, their explicit response to four shortcomings of existing datastore implementations in facilitating this kind of iterative and visual knowledge production. As a point of departure, "System and Method" points to current systems' inability to accommodate large datasets in the first place (scale); their inability to quickly access such data (latency); their inability to consolidate logically related but formally (that is, with respect to data format) heterogeneous datasets (silo-ing); and their inability to preserve the original form of those data.

Palantir implements new layers of abstraction and functionality using the operativity of existing distributed datastore technologies, including the capabilities of distributed NoSQL database systems (key-value stores in particular) like Apache Cassandra and Apache HBase.[39] The remainder of this chapter will consider three such "layers" of the Palantir platform: underlying distributed datastore technologies; Palantir's "key-value driver"; and their dynamic ontology feature. Each layer, in transforming the operational possibilities of the layer that precedes it, contributes distinctive techniques and processes that form part of the total internal image, the machinic address space of a Palantir deployment, of which the user interface is, ultimately, only an aspectual reduction.

---

[38] Geoffrey Stowe, Chris Fischer, Paul George, Eli Bingham, and Rosco Hill. System and method for investigating large amounts of data. US Patent 8,799,240, filed June 23, 2011, and issued August 5, 2014.
[39] Ari Gesher, "AtlasDB: Transactions for Distributed Key-Value Stores (Part I)," *The Palantir Blog*, https://palantir.com/2014/06/atlasdb-transactions-for-distributed-key-value-stores-part-i. (Page removed.)

4. Mapping and messaging: the logic of dispersion

This machinic address space is, however, often a composite of the address space of many different machines. Distributed technologies have been the information technology industry's primary response to the new challenges of storing more data and making it available to more users more rapidly: "the data center is the computer."[40] The intelligence of organizations, articulated with their capacity for managing growing numbers digital traces, becomes a feature that is increasingly restricted to those who have the capital and expertise to leverage these new distributed technologies. Within this landscape, the Palantir platform, like many other solutions, was developed as a proprietary system that adapts and builds on existing open source technologies. A 2014 post on the Palantir blog foregrounds scale and latency as the prime motivations in a move to adopt a distributed NoSQL datastore for their platform.[41] Indeed, the "System and Method" patent presumes the usage of a distributed key-value store as the foundation of a high-volume, low-latency datastore, but the patent does not cover in depth how distributed datastores improve the availability of data objects in the enterprise setting. Though these datastores are not Palantir creations – the invention described in "System and Method" is a "driver" for a key-value store, a "layer on top" – the operations of the distributed key-value store are central to Palantir's logic and functionality. In particular, "System and Method" cites two distributed NoSQL databases – Apache's Cassandra and HBase – as potential systems for the implementation of the driver described in the patent. This section will focus on technical documentation for Apache Cassandra and its forerunner, Amazon's Dynamo database management system, to further specify the mnemonic logic of distributed datastores and the Palantir platform. This logic emerges from the major design themes surrounding such systems, in particular their handling of the problems of mapping and messaging.

The discussion of Agre above underscores the general tension between abstraction and implementation that characterizes digital systems, a tension which distributed datastores recast in more specific terms. The unique design of a distributed datastore is the product of how the system balances two imperatives. The datastore should have an interface layer that presents users and applications interact with records as though they were drawn from a single, uniform data repository. At the same time, the availability and durability of the datastore relies on an underlying fragmentation and duplication, where subsets of data are stored on different machines (nodes) and multiple copies of data objects may exist. More concisely, the character of a distributed datastore results from how its computational subprocesses approach *mapping* and *messaging*. Mapping refers to the logic of the placement of data objects throughout the cluster, a logic also referred to as the distribution schema. Alongside this logic, there must be some messaging protocol for the retrieval of those data objects in response to user requests, as well as the exchange of metadata to update the operational status of nodes, reconcile conflicting versions of records, and other 'maintenance' operations. The three following subsections examine Cassandra (an explicit example in the "System and Method" patent) and Dynamo (often considered to be Cassandra's parent project) as suggestive approaches to these linked problems of mapping and messaging in distributed datastores like Palantir's.

---

[40] David A. Patterson, "The Data Center is the Computer," Communications of the ACM 51, no. 1 (January 2008): 105.
[41] Palantir Technologies, "AtlasDB."

Distribution schema

Consider an initial set of data objects to be distributed over a group of machines. The first problem of distributed datastores is that of partitioning. Different data objects or records are stored on different machines, in the hope of improving availability – horizontal scaling – but how can a designer ensure that these objects can both be stored and retrieved in an efficient manner? How can the system locate any given record? The answers to these questions constitute the system's distribution schema. Both Cassandra and Dynamo rely on a technique called consistent hashing.[42, 43] In computer science, a hash function maps data of arbitrary size onto data of a fixed size; this creates a mathematical signature of fixed length – an index – from some other bit of data, no matter its size. With a datastore, what is "hashed" is the unique key[44] of a data object. In consistent hashing, the upper and lower bounds of the hash function are fixed, such that the index for any given data will fall within a defined range. This range is then divided up amongst the N nodes of the Cassandra or Dynamo instance, each of which is responsible for storing a particular range of keys and their associated values; each node is the "coordinator" or "master" of a range of keys. By joining the upper and lower bound of this range the keyspace can be visualized as a ring, with the sub-ranges mastered by each node (A, B, C,…) occupying a length of arc along the circumference (Figure 1). In Amazon's "Dynamo" paper, DeCandia et al. claim that a particular advantage of consistent hashing is that it eases incremental scaling; the removal or addition of a node need only affect the nodes (and key ranges) immediately adjacent to that position on the ring.

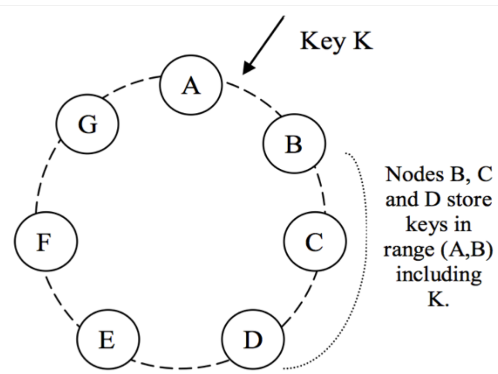

Figure 2.1: Visualizing the "ring" of key partitions in a consistent hashing scheme. Illustration from DeCandia et al., "Dynamo."

Each Dynamo or Cassandra node, then, maintains its own local map of the cluster, a table that links hashed key ranges with the routing information for the nodes that master that range. If a client requests a key and the original node is the master of that key, it returns the associated value for that key. But if that key is located elsewhere, the originating node directly routes client requests for that key to the responsible node. In this respect, the Dynamo paper describes the

[42] Avinash Lakshman and Prashant Malik, "Cassandra - A Decentralized Structured Storage System," *ACM SIGOPS Operating Systems Review* 44, no. 2 (2010): 35-40.
[43] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels, "Dynamo: Amazon's Highly Available Key-value Store," *ACM SIGOPS Operating Systems Review* 41, no. 6 (2007): 205-220.
[44] A key is a unique identifier for a data object, associated with a value (or set of values). For instance, in a system working with personnel records, a social security number could serve as the key, and other attributes such as name, address, phone number, and department, might be associated values.

system as “a zero-hop DHT [distributed hash table], where each node maintains enough routing information locally to route a request to the appropriate node directly.”[45] By way of contrast, the paper notes two other distributed hash table systems, Chord and Pastry, which rely on key-based, peer-to-peer routing schemes.[46] In these schemes, the same consistent hashing function generates node identifiers (randomly distributed throughout the “identifier circle”) along with key indices. While Dynamo and Cassandra distribute keys according to predetermined ranges, Chord and Pastry assign a key based on the mathematical proximity of its index to a node identifier. When a node receives a request for a key it is not responsible for, it will forward the request on to a neighboring node whose ID is “closer” to that key – in effect, passing the request “left” or “right” along the identifier circle. This approach is useful in that a given node only has to have limited information about the network topology – nodes do not maintain a routing table for the entire system – but a node can still “discover” a key through its neighbors.[47] On the other hand, this process of discovery may increase the traffic and latency in the system, as the requests are passed from peer to peer in search of the master node for that key. Dynamo, however, anticipates a use-case where the network topology is defined in advance, and is of a size (a few hundred nodes) where it would not be onerous for each node to maintain its own complete, local copy of the routing table.[48]

Replication and Consistency

After the initial partitioning in a Cassandra or Dynamo cluster, each node has certain data objects it is responsible for, along with an exhaustive map (stored locally) of the locations of the remaining keys in the system. Both Cassandra and Dynamo take advantage of their distributed character to replicate data, improving fault tolerance by maintaining multiple copies of data objects. In such systems, one node serves as the master or coordinator for a given key (record, data object) or key range. This master node then ‘pushes’ replicas to a predetermined number (the replication factor) of subsidiary nodes. These nodes, in turn, are coordinators for other key ranges and perform the same procedure for their respective keys. The master node and the replicating nodes together form a *preference list* for a particular key or key range. Each node maintains such a list, allowing for system access to that key range in the event the master node is unavailable. In cluster with a replication factor greater than 1, the “map” mentioned above will include alternative locations – the preference lists – for the each of the defined key ranges. Depending on the state of the distributed system as a whole, the “health” of its constituent nodes,[49] a given node’s map of the data should change, based on the preference list, in response to the failure of other nodes.

Maintaining multiple copies of data objects within the cluster poses the problem of ensuring all copies are consistent with one another. In some cases, updates may not propagate through the system before new client requests arrive. In others, multiple updates to the same

[45] DeCandia et al., “Dynamo,” 209.
[46] Rowstron and Duschel, “Scalable, decentralized object location and routing for large-scale peer-to-peer systems,” in *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, (Berlin/Heidelberg: Springer, 2001), 329-350; Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” *ACM SIGCOMM Computer Communication Review* 31, no. 4 (2001): 149-160.
[47] Chord, as described in Stoica et al., also assigns each node a limited number of links to nodes “across” the circle (“fingers”) to avoid all searches being linear searches.
[48] DeCandia et al., “Dynamo,” 208.
[49] Ibid., 211.

object may occur at around the same time, and the system must resolve these conflicting versions before propagating them. Solutions to these problems that maintain *strong consistency* in a database – that is, a guarantee that clients will always receive the up-to-date, canonical version of the data objects – reduce availability. For instance, a system that receives an update may "lock" that object and temporarily block subsequent write access to the data object until updates can be copied to the relevant nodes – that is, until the transaction has been committed. The system will then release the lock and permit the next transaction, ensuring that each transaction on a key is performed completely before permitting the next one.

A high availability system, however, that seeks to provide fast write access even in the event of server failures and network disruptions may relax such guarantees. Dynamo, for instance, is designed with the expectation that "customers should be able to view and add items to their shopping cart even if disks are failing, network routes are flapping, or data centers are being destroyed by tornadoes."[50] To accommodate such contingencies, Dynamo implements what is called *eventual consistency*, and permits continuous access (especially write access) while updates propagate in the background. While this approach improves the responsiveness of the datastore, it creates a problem with clients receiving out-of-date or conflicting versions of data objects.[51] In a use case like the Amazon shopping cart or a social networking profile, such conflicts may be a tolerable part of the customer experience. For instance, each user has the opportunity to review the contents of their cart before finalizing the ultimate purchase. In other contexts – for instance, where data objects are used in analytical processes – such uncertainty may not be acceptable, and stronger consistency guarantees are required. In addition to consistency, authoritative records of the moment and origin of data objects may be required for confidence in analytical products or for audit purposes. The Palantir platform addresses these issues by implementing transactions (including the key "locking" mentioned above) in an intermediate layer between applications and the eventually consistent key-value store.[52] The AtlasDB layer, running on a separate server for transactions, maintains a current list of locked keys whose updates are still making their way through the data store. A new write transaction on any of these keys client application would be forced to wait until any previous transactions had completed. But rather than the application waiting for confirmation of the transaction from the data store directly, the AtlasDB layer handles this processing, reducing the write overhead of the key-value store and maintaining strong consistency alongside its scalability and high read availability.

Key-value stores that do not implement a transaction layer may use other approaches for versioning and consistency. Dynamo, for instance, uses vector clocks to create causal version histories for each object. In this way, each data object has an ordered list of its update history, which includes an ID for the node that originated the update and a counter. With this, a system can decide if (and how) to reconcile conflicting versions based on their causal relationships.[53] Dynamo also uses a consistency protocol where a minimum number of nodes (i.e. a quorum) can be set as a threshold for declaring a read or write operation successful. For a read request, for instance, a coordinating node would gather existing versions of the data object from the N highest ranked nodes in the preference list. If all the versions are identical, the read is considered

[50] DeCandia et al., "Dynamo," 205.
[51] Ibid., 207
[52] Palantir Technologies, "AtlasDB."
[53] DeCandia et al., "Dynamo," 211.

complete and successful. If there are multiple, causally unrelated versions, these divergent copies are returned to the client, then reconciled and written back to the data store.[54]

Membership and Failure Detection

The preceding subsections discussed two linked procedures, consistent hashing and preference lists, for the algorithmic distribution (and replication) of data objects in Cassandra and Dynamo. A common logic coordinates the storage and retrieval of data, but this coordination occurs between the nodes as peers. One machine does not maintain and disseminate a canonical map associating machines and key ranges; rather, this map is constantly updated on and by the constituent nodes. Establishing the fact of nodes' common participation in a particular Cassandra or Dynamo instance (without recourse to a dedicated agent for coordination) is the task of the membership module. Membership subprocesses intersect with the keyspace partition logic to produce and maintain the contingent (but related) operational maps on each node that, together, define a running Cassandra or Dynamo instance.

Since the cluster has no centralized system for reporting these updates, membership changes must be communicated from peer to peer. The membership modules in these two systems utilize a gossip protocol to relay updates to each node as new nodes are added and failed or inactive nodes are removed from the membership roster. Gossip protocols, inspired by the "epidemic mechanisms" in the spread of rumors and disease, aim to efficiently propagate information throughout the distributed system. At defined intervals, such protocols call for nodes to choose "gossip partners" at random from the cluster and communicate any new information or "hot rumors."[55] For instance, in the case of Dynamo, "[e]ach node contacts a peer at random every second and the two nodes efficiently reconcile their persisted membership change histories"[56] and each node "actively gossips the full routing table,"[57] making these modifications to cluster membership part of the ambient traffic that maintains the coherence of the system. Gossip mechanisms may not rely only on random contacts. A limited set of nodes are defined in advance as seeds for these gossip exchanges and made known to all nodes, and each node reconciles its membership information with a seed at defined intervals alongside the random gossip exchange. In a large cluster, an update originating from a single site (out of hundreds) may not reach all sites within an acceptable number of gossip cycles; this seeding mechanism reduces the likelihood of temporary logical partitions (sets of nodes that are "unaware" of the existence of a set of other nodes) within the cluster.[58]

The timely and efficient propagation of explicit membership information presents one challenge for the design of distributed systems. Alongside, tracking intermittent unavailability (i.e. failure) of member nodes becomes onerous as clusters increase in size. Human monitoring of individual machine statuses would be near impossible in common use cases. Instead, failure detection processes running on the nodes themselves track and transmit information regarding unresponsive members. With Dynamo, "temporary node failures are detected by individual nodes when they fail to communicate with others (while forwarding requests)."[59] For instance, if

[54] Ibid., 212.
[55] Alan Demers, Dan Greene, Carl Huster, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dane Swinehart, and Doug Terry, "Epidemic Algorithms for Replicated Database Maintenance," *ACM SIGOPS Operating Systems Review* 22, no. 1 (1988): 8-32.
[56] DeCandia et al., "Dynamo," 212.
[57] Ibid., 218.
[58] Ibid., 213.
[59] Ibid.

node A sends a message to node B and receives no response within a defined period, then node records node B as down and uses other hosts that map to B's partitions to service its requests. Node A will periodically retry B to see if it has recovered, but this local failure detection process relies on client traffic to drive the failure detection process. Dynamo's designers claim that "[i]n the absence of client requests to drive traffic between nodes, neither node really needs to know whether the other is reachable and responsive," an approach that may reduce the computational and network overhead associated with maintaining a global account of temporary node failures.[60]

While the Dynamo paper describes a Boolean failure detection model that marks a node as up or down when forwarded requests go unacknowledged, Cassandra uses an accrual failure model to reduce the likelihood (and costs) of routing requests to unavailable nodes.[61] Cassandra's failure detection implementation assigns a value Φ to each node, with this value indicating a 'level of suspicion' that a given node is unavailable. This value is calculated based on the distribution of reply times for gossip messages from a given node over a moving time window. The authors of the Cassandra paper claim this statistical method is a more flexible approach to failure detection and can easily adapt to different network and load conditions.[62] Across different systems and deployments, the degree and logic of the uncertainty regarding the status of nodes varies according to the failure detection model in use. That is, for the sake of efficiency, any given node tolerates some amount of indeterminacy in its knowledge of the cluster as a whole.

These examples of distributed datastores, Cassandra and Dynamo, can be thought of as a collection of computers with more or less extensive and/or synchronized maps indicating which nodes "own" certain data objects. These maps that are constantly amended in response to node failures and network partitions, and they can be seen the result of the ambient communication and background traffic that goes into the cluster's maintenance of its own coherence. The techniques employed to create a system like Dynamo (which was designed to support users' views of their Amazon.com shopping carts), a system that is responsive around the clock to millions of concurrent users, are suggestive of the technoepistemological character of not just Palantir, but a range of data warehousing and analysis systems and, indeed, many contemporary Internet applications.

No matter which node a client is connected to, that client has access to a global view of the dataset, not just the fragment stored locally on that node. A distributed datastore projects a single, uniform data environment, wherein a client can (at least potentially, depending on their permissions) access any object from the entire dataset without referring to the specific location of that data object or the cluster's network topology. The datastore hides the details of this implementation from clients, rendering invisible the far-flung servers, updates, load balancing, crashes and recoveries.[63] The symbolic layer of data records is constantly knit and patched

---

[60] Ibid.

[61] Lakshman and Malik attribute the Cassandra model to Naohiro Hayashibara, Xavier Defago, Rami Yared, and Takuya Katayama, "The φ accrual failure detector," in *Proceedings of the 23rd IEEE International Symposium on Reliable Distributed Systems, 2004,* 66-78.

[62] Lakshman and Malik, "Cassandra," 37.

[63] For a more concrete sense of scale, the Dynamo test instances described in their paper were running on a few hundred nodes. The 2009 Cassandra paper describes Facebook's experience with a 150 node cluster working with 50+ TB of data, providing the ability to search message inboxes by terms or by interactions. Latency results for the Cassandra searches were on the order of tens of milliseconds. According to a 2010 technote from Facebook engineer Kannan Muthukkaruppan, Facebook eventually moved from Cassandra to HBase as a distributed datastore for their message functionality, due to consistency issues.

together by systems and procedures that manage and renew the logical interrelation of the autonomous nodes, and embodies a global schema that presents these interrelated elements as a single functional entity. Client applications interact with an application-level network, also known as an overlay network, that abstracts from a substrate network topology to create a virtual links and a logical topology.[64] With the consistent hashing approach in Cassandra and Dynamo, the address space at the application-level may be understood as a ring (see figure 1 above), and a direct link (a chord) may be created between each node on the ring's circumference. The architecture of the distributed datastore, then, is in the procedures that establish and the messaging processes that maintain the relations of the nodes, presenting dispersed and replicated fragments of a dataset as a single consistent and uniform set of objects with associated attributes. An abstract, logical layer emerges as the result of a continuous (and tenuous) networked coherence. The primary unit in this configuration, however, is the key-value pair, a unit that sits at some distant from the investigations centered on the arrangement of real world objects seen in chapter 1. Taken on its own, a Cassandra or Dynamo instance does not easily accommodate the semantic requirements of human analysts—for this, an additional layer is needed to translate these flexible but simple digital records into structures more fully representative of human understandings of the application domain.

5. "Another level of indirection": The Palantir key-value driver

The preceding sections described in detail the mnemonic logic of the distributed datastores, systems that provide large amounts of data storage with high availability and rapid access to the data objects therein. The interactive possibilities of such a system, however, are not "intuitive" for human users conducting analysis. This section describes a subsequent, intervening layer for referencing key-value pairs, one that coordinates and composes them as more familiar data objects which can then be addressed as such by the application layers (e.g. from the graphical user interface). Generally speaking, the Palantir key-value driver can be described as an instance of a fundamental approach in computer engineering. Classic theories of artifice and design emphasize attention to the interface between the "inner" world of an artifact and its "outer" environment. Design, in this view, is a matter of configuring a physical system to align its possibilities – its set of possible states – with an abstract definition of a task or task environment.[65] With computational systems, the notion of abstraction layers denotes the hierarchical relationship among different aspects of the digital artifact, from semiconductor to software, and how each layer creates new functional possibilities by organizing and combining the primitive operations made possible by the layer "below." Digital systems highlight how artifacts redistribute and transform possibilities offered by some other artifact to provide new functionality in a task environment, and, in the case of key-value driver, the goal of this redistribution is to bring the logic of the datastore "closer" to the logic of human interaction and investigation.

Much of computer engineering centers on novel repackaging of the functionality offered by some existing system or set of systems. Cassandra and Dynamo, as we have seen, combine

[64] John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole, Jr., "Overcast: reliable multicasting on an overlay network," *Proceedings of the 4th conference on Symposium on Operating System Design & Implementation* (2000).
[65] Simon, *The Sciences of the Artificial*, 113, 121.

local storage and networked communication to present a uniform logical layer to applications and end users, and they make it possible for those users and their applications to access individual data records without regard for which actual system those records may be stored on. In Dynamo, client applications may simply "get" and "put" key-value pairs and need not address (or even understand) the mechanisms for their distribution and retrieval across the cluster. The application programing interface provided by the datastore functions as a "wrapper" for the constituents of the distributed system with basic operations more appropriate for the abstracted task. The invention described in "System and Method" patent, what Palantir refers to in other contexts as their "key-value driver" reiterates this 'wrapper' logic, the logic of abstraction. The key-value driver acts as bi-directional interpretation layer (see figure 2) for the underlying datastore, providing an interface, a set of functional possibilities, more closely attuned to the archetypical investigative contexts Palantir highlights.

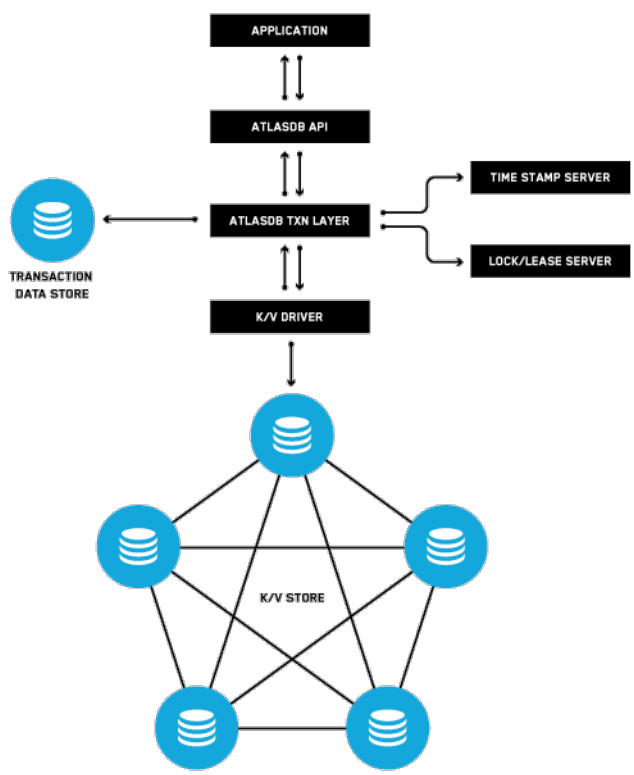

Figure 2.2: Diagram of relationships among components of the Palantir datastore, including key-value store, driver, transaction layer, and application. From Palantir Technologies, "AtlasDB."

While key-value datastores are often cast as ameliorating the problems of scale and latency with large datasets, the problem of fine-grained search remains. In their default implementations, Cassandra and Dynamo only provide "primary key" access (e.g. the operations get(key), put(key,value)/insert(key,value), and delete(key)) to data objects when the key is known *in advance*. In the case of the online shopping cart, for instance, a user logs in with their unique ID (say, an e-mail address) which functions as the primary key for their shopping cart data object. In this use case, both other keys (other users' shopping carts) and the values (items in the cart) are of little concern for identification and retrieval of the necessary record. According to Palantir, however, a datastore that only offers this kind of primary key access falls short in investigative use cases where, as the "System and Method" patent mentions, the data object in question may not be known in advance. The patent presents the example of an investigator, searching through an organization's archived e-mail messages, who will need to filter the messages first by sender and then by date.[66] How can one model and store data in a key-value system to support this kind of search functionality? What application logic is needed make these other attributes of the logical data entity accessible at the level of the datastore?

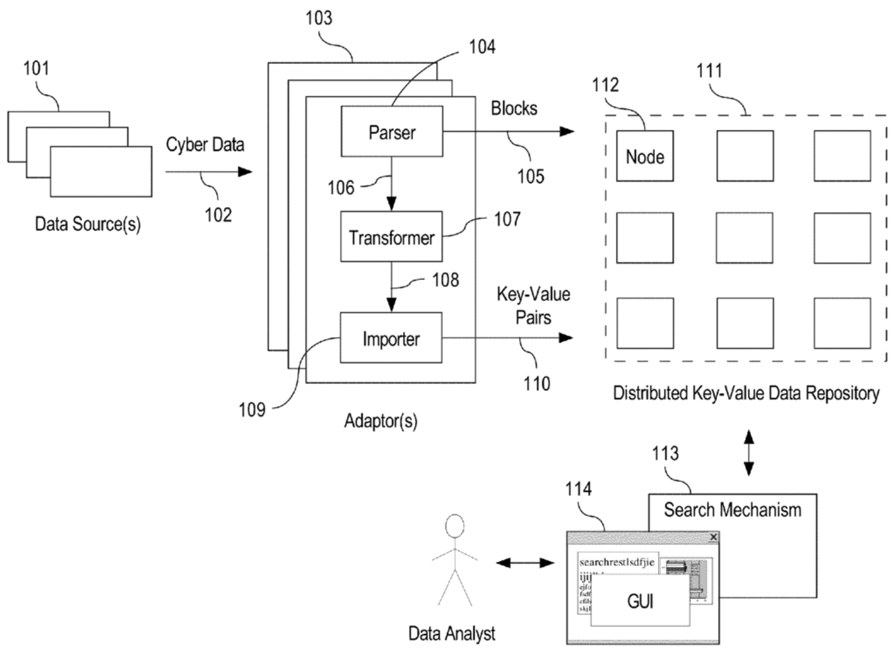

Figure 2.3: Diagram representing systems for importing data into Palantir datastore in Stowe et al., "System and Method."

The Palantir key-value driver implements this search functionality through the use of interrelated key-value families. Remaining with the e-mail message example, the first key-value family would contain senders' e-mail addresses as keys. The associated values would consist of unique identifiers for the individual e-mail messages sent by that user – in the patent, these identifiers are a combination of the sender address and a timestamp (seconds elapsed since the beginning of January 1, 1970 GMT). Part of the definition of that key-value family would be that associated values serves as keys (or point) to pairs in a second key-value family. This second set of identifiers, in turn, become the keys for in a second key-value family. The values in this family identify data blocks where the e-mail message itself is stored.

[66] Stowe et al., "System and method," Section 2.4.3.

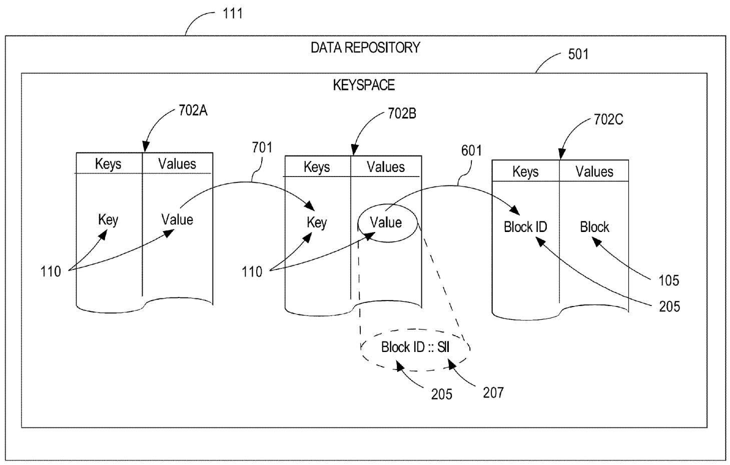

Figure 2.4: Representation of interrelated key-value families in Stowe et al., "System and Method."

As data objects, e-mail messages have multiple relevant attributes (sender, recipient, date/time sent, subject). Though the datastore (the Dynamo or Cassandra instance) sees the keys as a one-dimensional pile of data objects, the key-value driver accounts for associated attributes, parsed from the incoming data, and integrates this information into the generation of keys and values. Figure 6 presents an example of this integration using the syntax of alphanumeric strings from their e-mail message example:

Family ID: Users	
Key	Value
john.smith@example.com	john.smith@example.com_1282589884, john.smith@example.com_1678342884

Family ID: SentMsgs	
Key	Value
john.smith@example.com_1282589884	[Block ID for stored message]

Figure 2.5: Example of interrelated key-value families for an e-mail message repository in Stowe et al., "System and Method."

A request for the key john.smith@example.com would return the two time-stamped values, with the time-stamped values serving as a unique identifier for a particular sent e-mail message. Using these values, then, the driver requests key-value pairs in a second key-value family for sent messages. The values in the second family could be data block identifiers that point to the location of the original message text in the datastore.

For the underlying Cassandra (or other key-value system) instance, the keyspace may remain undifferentiated with respect to these families, and the constituent keys distributed across the cluster according desired parameters for latency, replication, and other factors. The Palantir key-value driver, however, maintains links between values of the first key-value family (item 702A in figure 4) and the keys for the second key-value family (702B), as well as between the values of the second key-value family and the third (702C) family linking identifiers to specific data blocks. In essence, the key-value driver overlays a second mapping on the original distribution schema for the keyspace, creating distinct but linked key-value pairs that refer to pre-defined aspects of the data object. These keys may be stored anywhere on the cluster, but the key-value driver maintains the coherence of the record such that it can be addressed based on any of the elements that have been mapped to a key-value family. The key-value store, as mentioned above, coordinates maps between independent and heterogeneous nodes to rapidly locate key-value pairs. The Palantir key-value driver, as a layer on top of the datastore, maps key-value pairs (regardless of location) to create relationships that can reflect human analytical understandings of the components of a data object. In this respect, the key-value driver functions as an *interpretation layer*, mediating between the fast (but less logically sophisticated) distributed key-value store and the more complex understandings of data desired for higher level applications and analysis.

Adaptors

The above discussion of the keyspace, key-value families, and key-value pairs outlines the abstract logic of Palantir's "System and Method" patent and how its procedures assemble primitive data objects as related attributes of the logical data entity, facilitating operations with arguments beyond the primary key. The key-value driver extends the basic mnemonic logic of the datastore, providing a second, logical map of related attributes articulated with the datastore's (locational) partition map. However, the sources, "log files, e-mail message spools, transaction logs, call data records," circulate in formats that may not be easily convertible to key-value pairs. Translating the mnemonic logic of various data sources to provide content for a separate analytical datastore is a task for the adaptors. The "System and Method" patent claims that the resulting system is "agnostic" as to these different data sources and formats, and it can "ingest virtually any type of structured data."[67]

```
nasa_19950801.tsv
mac19.ils.uec.ac.jp	-	807257019	GET	/images/WORLD-logosmall.gif	200	669
dd10-046.compuserve.com	-	807257020	GET	/images/launch-logo.gif	200	1713
server.indo.net.id	-	807257039	GET	/shuttle/missions/sts-37/sts-37-patch-small.gif	200	10371
dd08-050.compuserve.com	-	807257043	GET	/history/apollo/apollo-13/movies/apo13damage.mpg	200	297851

mtv-pm0-ip4.halcyon.com	-	807257053	GET	/shuttle/countdown/liftoff.html	200	5220
mac19.ils.uec.ac.jp	-	807257063	GET	/whats-new.html	200	18936
osc_pc3.79.242.202.in-addr.arpa	-	807257066	GET	/shuttle/missions/sts-71/images/index71.gif	200	337360

mac19.ils.uec.ac.jp	-	807257067	GET	/images/whatsnew.gif	200	651
in2pc2.med.niigata-u.ac.jp	-	807257078	GET	/	304	0
in2pc2.med.niigata-u.ac.jp	-	807257080	GET	/images/ksclogo-medium.gif	304	0
in2pc2.med.niigata-u.ac.jp	-	807257080	GET	/images/NASA-logosmall.gif	304	0
in2pc2.med.niigata-u.ac.jp	-	807257080	GET	/images/MOSAIC-logosmall.gif	304	0
in2pc2.med.niigata-u.ac.jp	-	807257080	GET	/images/USA-logosmall.gif	304	0
in2pc2.med.niigata-u.ac.jp	-	807257081	GET	/images/WORLD-logosmall.gif	304	0
van13383.direct.ca	-	807257092	GET	/software/winvn/winvn.html	200	9866
van13383.direct.ca	-	807257094	GET	/software/winvn/winvn.gif	200	25218
van13383.direct.ca	-	807257094	GET	/images/construct.gif	200	1414
van13383.direct.ca	-	807257094	GET	/software/winvn/bluemarb.gif	200	4441
mtv-pm0-ip4.halcyon.com	-	807257094	GET	/htbin/cdt_clock.pl	200	543
c3.reach.net	-	807257098	GET	/htbin/cdt_main.pl	200	3714
mtv-pm0-ip4.halcyon.com	-	807257101	GET	/images/NASA-logosmall.gif	200	786
c3.reach.net	-	807257101	GET	/shuttle/countdown/images/countclock.gif	200	13994
van13383.direct.ca	-	807257102	GET	/software/winvn/wvsmall.gif	200	13372
c3.reach.net	-	807257105	GET	/images/NASA-logosmall.gif	200	786
van13383.direct.ca	-	807257110	GET	/images/KSC-logosmall.gif	200	1204
van13383.direct.ca	-	807257115	GET	/images/MOSAIC-logosmall.gif	200	363
in2pc2.med.niigata-u.ac.jp	-	807257121	GET	/images/shuttle-patch-small.gif	200	4179
dd10-046.compuserve.com	-	807257124	GET	/history/history.html	200	1602
van13383.direct.ca	-	807257126	GET	/images/USA-logosmall.gif	200	234
dd10-046.compuserve.com	-	807257127	GET	/history/apollo/images/apollo-small.gif	200	9630
dd10-046.compuserve.com	-	807257127	GET	/images/NASA-logosmall.gif	200	786
```

Figure 2.6: Apache web server access logs in tab-separated value format. Retrieved from http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html (Author screenshot)

Adaptors, as described in "System and Method," are modules designed to render character data from "virtually any" source as interrelated key-value families and key-value pairs within the distributed datastore.[68] Web log data and e-mail message spools are certainly not unreadable – an experienced system administrator would be well acquainted with their processes

[67] Stowe et al., "System and method," Section 1.0
[68] Ibid.

of generation and resulting syntax. But the aggregation and representation of these data as discrete logical entities available for automated search is, presumably, a much more desirable arrangement for the investigative contexts.

The adaptor module described in "System and Method" is a data processing pipeline composed of two sequential components (with an optional module between, see figure 8). First among them is the parser, whose function is to divide the input data stream into discrete, logical sections based on defined criteria, to "identify logical data entities in the input data" to pass to the importer for inclusion in the datastore. A "logical data entity may be an email message, log file entry, a call data record, a netflow record, or *any other logical entity of data* [emphasis mine]."[69] For instance, with the Apache access log in Figure 7, the first step would be to select each new line in the log file (which begins with IPv4 address that originated the request) and render it as a single logical entity – one website request. Then, based on other syntactical markers (e.g. the tabs in the Apache log), the parser would extract a time and date of access, the type of request, the resource requested, the protocol used, and the resulting status code of the request (success, redirection, client error or server error).[70] Various strategies can be employed to parse input data streams, but this basic implementation relies on the predictable syntax of such a log file (byte count, delimiting characters).

A version of the original data format is translated for inclusion in the data store, but the original or "raw" data is not discarded. Rather, the source data in its original format is also indexed and stored (for later "contextual analysis" if needed). A linked key-value pair might, for instance, refer to the line number in the log file associated from which a data object was derived. (See item 203 in Figure 9.) This parallel retention and indexing of data in its original format suggests that whatever relations are initially implemented by a specific Palantir instance and its system of key-value families are presumed to be only an approximation. With the capacious and scalable storage made available by cluster and cloud computer, there are fewer obstacles to archiving source data and linking them to the higher order objects (that still reside as key-value pairs) within the data repository. By holding this "raw" data in reserve, Palantir's mnemonic logic holds open an adjacent ground for investigation, not necessarily limited by the current data model. This arrangement suggests a modesty about data models and the ontologies they imply, and the retention of source data permits alternative modes of cognition with respect to the problem in question.

---

[69] Stowe et al., "System and method," Section 2.4.1

[70] "Log Files", Apache HTTP Server 2.2 Documentation. https://httpd.apache.org/docs/2.2/logs.html. Accessed 23 November 2015.

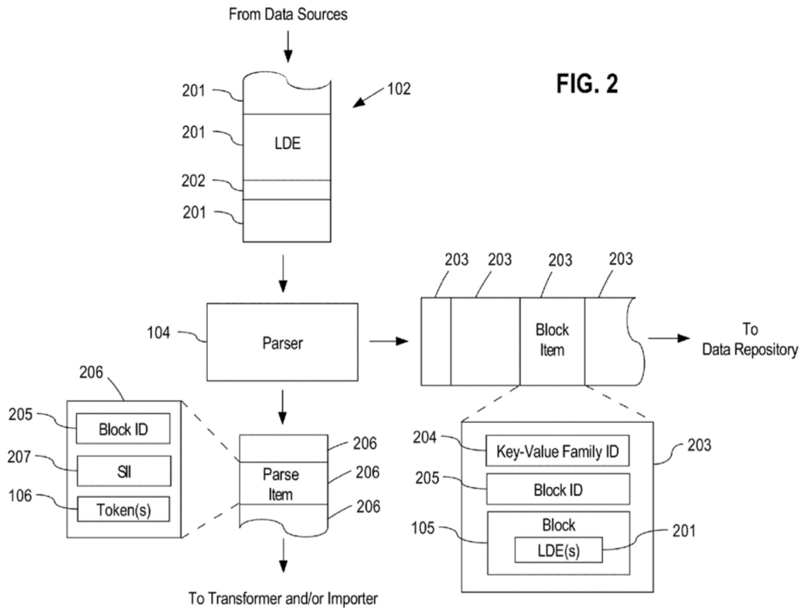

Figure 2.7: Detailed diagram of process for data extraction and importation described in Stowe et al., "System and Method."

Continuing with the pipeline, once the parser has generated the parse tokens – snippets of desired data extracted from the messy syntax of the original data stream, character data newly aligned with the logic and implementation of the datastore, an optional step of transformation may follow. In this step, parse tokens that may be nothing more than simple excisions from the original data stream may be converted to a standard format. The transformer module might, for instance, convert all string values to lowercase, convert various time/date formats into a canonical representation (seconds or milliseconds since an epoch), along with many other possibilities. Depending on the nature of the data and the analytical goals, the adaptor logic may not include a transformation step. But canonicalization, conversion, and other transformations allow for the encoding of extensive relations among key-value families and pairs – for instance, keys for sent e-mail messages are formatted in the style of "john.smith@example.com_1282589884" allows the key-value driver to easily parse out timestamp information for range queries within the key-value family for sent messages.

Following the parsing and (optional) transformation, the importer is then tasked with generating data objects (i.e. making put(key,value) requests) in the datastore based on those parse tokens. These key-value pairs (which, together, represent related attributes of the overall logical data entity) also index blocks of data in the original format for later retrieval of that initial content in concert with searches of the key-value store. The importer takes a "stream of parse items from the transformer… [which] may contain a data block identifier, snippet identifying information, and/or one or more parse tokens and/or one or more transformed parse tokens [. . .]." "How the importer forms and generates key-value pairs from the input stream of parse items will depend on the expected searches to be performed" – in other words, what "logical data entities" the organization has deemed to be relevant in the input data stream.[71]

[71] Stowe et al., "System and method," Section 2.4.3

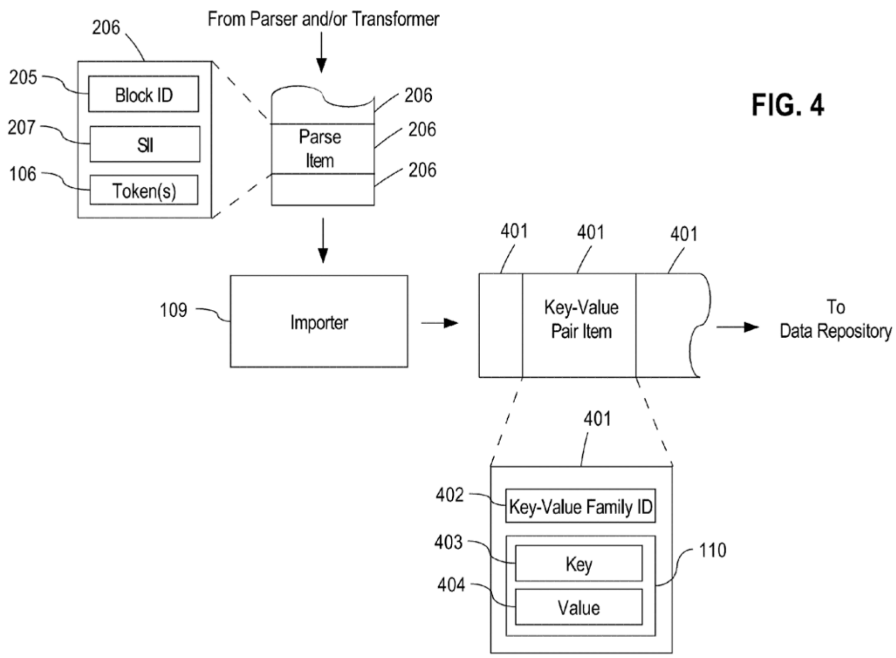

Figure 2.8: Diagram of final step of data conversion and ingestion as key-value pairs in Stowe et al., "System and Method."

Put otherwise, "how the importer forms and generates key-value pairs from the input stream of parse items will depend on the expected searches to be performed."[72] The specific deployment of the adaptor/importer determines what level of "depth," where along the chain of key-value pairs that point to one another that the reference to the specific data block will be stored, and will embed the appropriate syntax for recognizing and retrieving related keys from other key-value families (the underscore and timestamp with the emails, for example) when the first key (the username) is requested. These operations occur at the level of the Palantir key-value driver and its adapters. At the end of all of these processes, the results become arguments for simpler get() and put() API calls (operations) to the underlying Cassandra, Dynamo, or HBase distributed key-value datastores.

Broadly speaking, the Palantir key-value driver architecture facilitates a two-step information retrieval process using a series of separate maps. First, we need a key-value functionality that can locate data objects based on elements from set of defined attributes (sender, receiver, date, time, etc.). In turn, the key-value driver will also index and link the data object that stores the original contents of the data stream (the portion of the log file or the e-mail message itself) for further analysis. Finally, the underlying key-value store (Cassandra, Dynamo, or HBase instance) locates and returns the key-value pair from a specific physical machine in the cluster. The adaptors' preservation of original data implies that insight may result from a human analyst's consideration of the object renderings of source data and their original syntax. For human analysts, the data objects indicate something beyond their instantiation in the datastore. This ability to ask questions outside the dataset becomes crucial for other aspirations of the

[72] Ibid.

platform. The architecture parses and imports data sources such that they can be represented in the Graph as systems of nodes and edges for visuospatial reasoning and pattern recognition, and this mode the most prominent form of "reading" data suggested in their archive. But, if desired, the Palantir platform also holds in reserve the untransformed data blocks from sources for other forms of reading in parallel or on concert. The analyst's sense of the dimension of the problem need not be restricted to the flat space of the current database; their awareness of the causal chains that generate the data points, as well as other, heterogeneous data related to the phenomena in question may permit alternative analytical modes.

The design of the platform does not anticipate or facilitate a lone, superlative rendering of digital data for analysis – human or machinic. Rather, as described earlier, Palantir characterizes the platform as an *interpretation layer*. This "interpretation" could refer to the work of the adaptors, their extraction of signifying components of heterogeneous data sources, and the storage of those signifying traces as separate and searchable data objects (i.e. key-value pairs). According to "System and Method," adaptor functionality can be customized for any number of input data streams (some which may be continuous), for example "web access log data" and "e-mail server data" for the Los Angeles office as separate (or combined with) the New York office, or other database systems.[73] Interpretation, at the level of the adaptor and datastore logic, would appear as the automated translation of a collection of digital traces in one format to a second form more compatible with non-machinic sensibilities. Another level of interpretive work, however, occurs with respect to the sociotechnical phenomena those traces claim to describe, in the analyst's arrangement of minute, archaic traces to speak of those things "behind" the data: agents, actions, and motives, and the other macrotemporal phenomena of historical time.[74] The human-driven navigation of different (but internally consistent) views or aspects of the n-dimensional image, the movement Engelbart claims as a crucial part of the efficacy of computers in intelligence augmentation, extends beyond a single dataset in the Palantir environment. The key-value driver structures the simpler key-value store, allowing for the storage of data entities from multiple, heterogeneous sources within the same keyspace – in addition to the indexed data blocks of the source. This consolidation enables reciprocal operation between data sources, allowing search results from different datasets to be visualized in the same environment, and potentially providing new dimensionality to a given subset data. As an interpretation layer, Palantir withholds a complete, algorithmic adjudication of the relationship between source data and data objects and between different collections of data objects. In doing so, the platform holds open a space for the human analyst to modify not only the size, but the dimensionality of the data subset under scrutiny and the interpretive method – which need not be declared to proceed. The mnemonic logic of Palantir aims to make available different kinds of determinations for digital traces and data objects, permitting open-ended and inexact readings of interesting subsets of data objects.

6. Dynamic Ontology and Reprogrammable Organizations

The chapter thus far has traced two layered systems that enable the persistent storage, retrieval, and association of data objects – a value can be stored, called up, as well as refer to other data objects stored in memory. The *meaning* of any of these data objects, however, is not

[73] Stowe et al., "System and method," Section 2.4
[74] Ernst, "Media Archaeography," 241.

explicitly addressed by either the datastore (key-value store) or the key-value driver. The organizational and social functions of a database hinge on common and stable expectations of "correspondences between constructs inside the information system and in the real world."[75] The values, often alphanumeric strings, should be intelligible *as* some thing or another. These expectations pose the problem of stabilizing the meaning of data across an organization, both in the present and in the future. The "representational machinery with which to instantiate domain models in knowledge bases" is referred to as a database ontology.[76] Another layer of abstraction is added to overcome the "indifference to meaning" inherent in digital technology.[77] Ontologies are machine resident metadata that indicate the semantics of data objects and their reference to other data objects; an ontology is the extra set processes required to articulate the 'flat' logical and material space of digital archives with the "'deep' hermeneutic space" of society and culture.[78] Moreover, changes in the fundamental objects, properties, and relationships in an application domain present additional obstacles to the continued intelligibility of digital datastores – database users expect semantics to be not only clear and stable, but also relevant and up-to-date. And, as social action becomes increasingly inflected by digital technologies, the rapid emergence of new, heterogeneous, and often unstructured data sources draws continued attention to the inertia of database systems – including the ontologies – already in place. Techniques to enable the evolution of database schemas and ontologies are, in turn, conditioned by this more general tension between digital microprocesses in place and the shifting semantics of an organization and its world.

Modifications to the higher-level conceptual schema may require concomitant changes in the encoding procedures that translate these 'infological' attributes into machine-readable syntactical patterns.[79] (Recall, for instance, the combination of e-mail address, underscore, and fixed width integer timestamp to reference sent e-mail messages described in the sections above.) These challenges present another instance of the abstraction-implementation dialectic. For instance, in the Cassandra and Dynamo datastores described earlier this chapter, the internal schema of the key-value store only supports the assertion that a key K and a value V are associated, without any information about the meaning of this association. The key-value driver's implementation of key-value families permits the association of multiple key-value pairs and, if desired, a link to a source data block. However, a conceptual data schema must be mapped onto this internal schema using additional procedures to encode (and decode) notions like object types, types of relationships, class membership, and other logical constructs.[80] The Palantir platform responds to this tension with the introduction of a "dynamic ontology" coupled to their datastore. Its aim is to reduce the costs, in terms of the labor of reprogramming, of modifying this crucial abstraction.

---

75 William Kent, *Data and Reality* (2000), 1.
76 Tom Gruber, "Ontology," in *Encyclopedia of Database Systems*, eds. Ling Liu and M. Tamer Özsu, (New York: Springer 2009), 1964.
77 Ernst, "Media Archaeography," 245.
78 Ibid., 248.
79 Gary H. Sockut and Robert P. Goldberg, "Database Reorganization – Principles and Practice," *ACM Computing Surveys* 11 no. 4 (Dec. 1979), 371-395.
80 Alexander Borgida, Marco A. Casanova, and Alberto H.F. Laender, "Logical Database Design," in *Encyclopedia of Database Systems*, eds. Ling Liu and M. Tamer Özsu, (New York: Springer 2009), [pp?]

Agnostic Abstractions

In describing their platform, Palantir invites us to view their platform as an "operating system" for data analysis, presenting a "set of clean abstractions" for search, visualization, and other manipulations of data.[81] The introduction of new layers of abstraction typically increases the computational overhead in a system, as operations require extra computational 'steps' to return a result. As a consequence, many production databases (especially relational database systems, a frequent foil in discussions of NoSQL-inspired systems like Palantir) are designed with a closer correspondence between their conceptual data model and their internal schema. One frequently cited drawback is that subsequent changes are "typically extremely disruptive" and costly in terms of both computational and human resources.[82] Palantir's "operating system for analysis" paradigm calls for implementing the ontology as a set of definitions for object types, properties, and parsing of input data *separate* from data sources and their internal schemas. This separation of concerns permits the customization and modification of these types and mappings using Palantir's dynamic ontology tools. Palantir presenters are keen to remind that this is not a hard-coded system but open to straightforward modification by the client organization.[83]

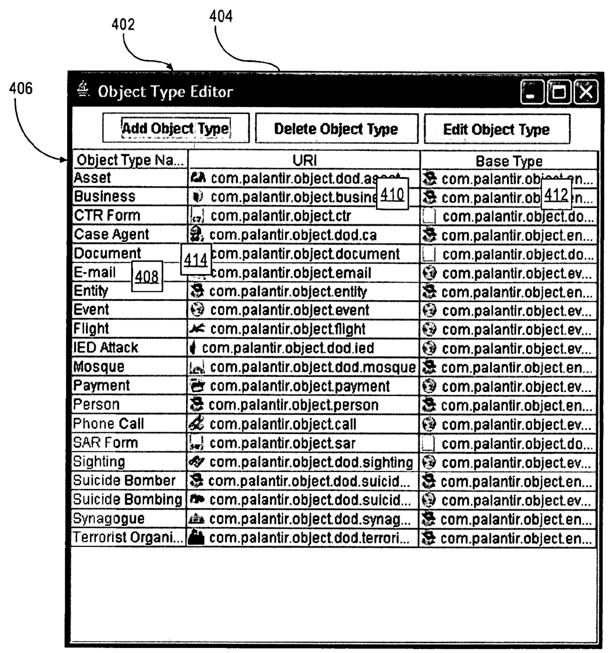

Figure 2.9: User interface screenshot of Object Type Editor from Jain et al., "Dynamic Ontology," showing relationship between object types, URI (universal resource identifiers), and base types.

The ontology's conceptual objects, "essentially typed containers for properties," are arranged in a hierarchal type system.[84] Administrative users can define object types and their properties using editors within the Palantir environment's GUI. Figure 9 above provides an example not

[81] Palantir Technologies, "Palantir: like an operating system for data analysis", http://palantir.com/2009/11/palantir-like-an-operating-system-for-data-analysis (accessed May 9, 2016. Page removed.)
[82] Akash Jain, Robert J. McGrew, and Nathan Gettings, Creating data in a datastore using a dynamic ontology, US Patent 7,962,495, filed Nov. 20, 2006, issued June 14, 2011. An extensive online bibliography of papers on schema evolution from Rahm and Bernstein can be found at http://se-pubs.dbs.uni-leipzig.de/.
[83] Palantir Technologies, "Dynamic Ontology."
[84] Palantir Technologies, "Palantir: like an operating system for data analysis."

only of the editor interface but also some potential object types (terrorist organization, mosque, suicide bomber, business, asset) in an intelligence context. Along with containing properties as textual object attributes (strings, integers, dates), Palantir objects can link to media in other binary formats (audio, video, and images), as well as free form text notes, as depicted in figure 10.[85]

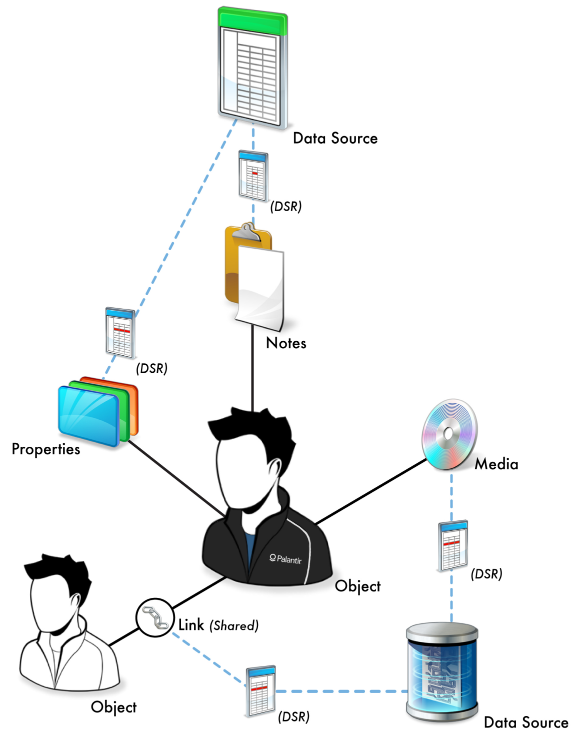

Figure 2.10: Diagram of relationships between objects, properties, and data sources with mediating DSRs (data source records). From: Palantir Technologies, "Palantir: like an operating system for data analysis."

The objects themselves, however, are the result of a process of composition from available data sources; data sources may *not* (and need not) be organized in advance according to a given ontology. A Palantir deployment, through its selected data importer plugins, reads data (which will become object properties) from those sources and, according to the definitions set in the ontology, "wraps" them up into PTObjects. The ontology remains ultimately "agnostic" to the underlying data sources (relational databases, servers generating access logs) that supply data to be copied into the higher-level PTObject containers.[86] Analytical applications interact with data sources through the mediation of the ontology layer, with clients making requests or calls to read and write *only* in terms of composite PTObjects.

Object types are also associated with parser definitions, instructions for the "glue code" that identifies syntactical patterns or other means to extract relevant information from data sources and prescribes how the system will transform and import that source data in the form of PTObjects. Similar to the adaptor modules described in the "System and Method" patent, the parsers provide for the characterization of actual data sources and the extraction of data to populate an ontology with a plurality of objects. One comparison given (again, with the operating system analogy in mind) is that of a block device driver – the user of a Windows, Mac,

---

[85] Ibid.
[86] Palantir Technologies, "Palantir: like an operating system for data analysis."

or other operating system only “sees” files regardless of whether the data is present on flash media, magnetic hard disk, an optical disk, or network storage.[87] New data sources (or storage media) would only require the design of new drivers or adaptors to hide the technical details of their operation and present the contents as familiar conceptual objects – a business, a person, a vehicle, or a cell phone.

Palantir’s approach to data sourcing provides two additional axes of differentiation within its mnemonic logic. Through the use of *tethered sourcing*, analysts can view the origins of the elements of a data object and assess their quality, and administrators can precisely control access to individual data points within those objects to address both security and privacy concerns.[88] PTObjects contain not only properties and relationships, but data source records (DSRs) accompany each attribute of relationship ascribed to an object. These records, as with the data block identifiers in the key-value driver, reveal the origin of the data that “supports” the property attributions that make up the object – which imported spreadsheet, log file, or database contributed information for the composition of this object.

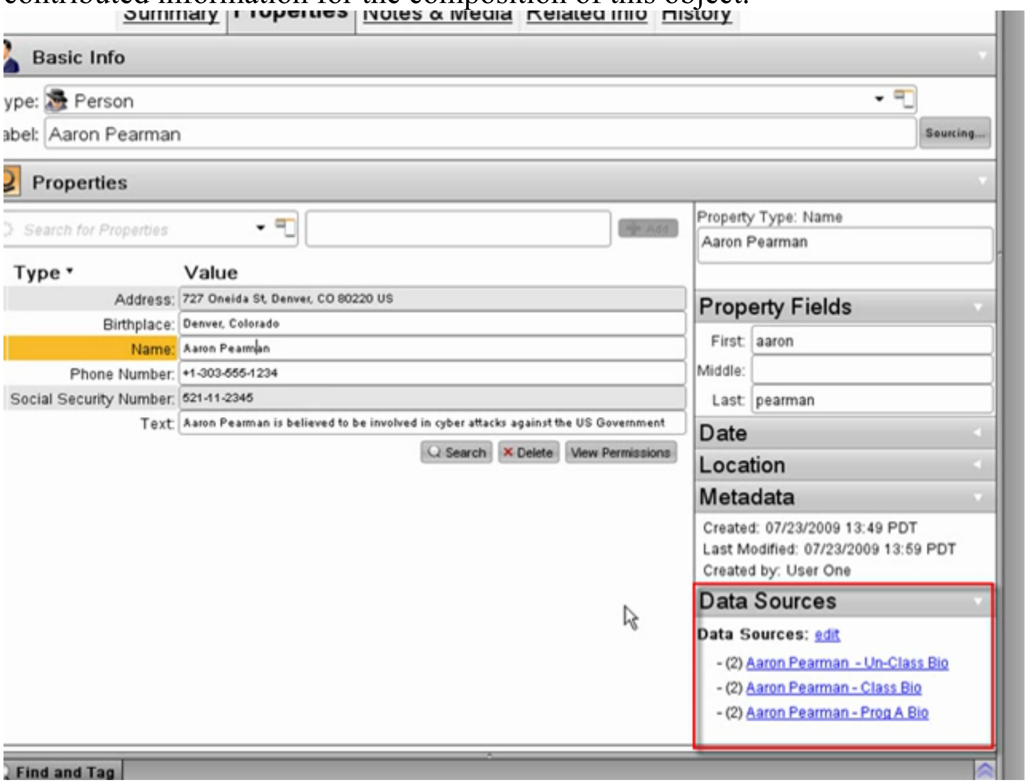

Figure 2.11: Highlight of data source for object properties in Palantir environment. Author screenshot from Palantir YouTube video, “Multi Level Security – Part 2.”

Users can “independently review the provenance and pedigree of data” as part of their investigation, as pictured in the dossier view in figure 11. Through the data source records associated with PTObjects, users (depending on access privileges) can review the “a complete history of the data” and “track additions, modifications, and deletions.”[89] Speaking to

---

[87] Ibid.

[88] Palantir Technologies, “A Core Commitment: Protecting Privacy and Civil Liberties,” https://www.palantir.com/wp-assets/wp-content/uploads/2012/06/ProtectingPrivacy_CivilLiberties_2012.pdf (accessed June 12, 2016.), 4.

[89] Ibid.

organizations and analysts, a Palantir white video presenter emphasizes the need to be able to trace an analytic product to its origins: "You need to know [the product's] lineage, its pedigree. This is the only way you can confidently state any assertion."[90] Domain expertise includes a skepticism toward data sources on the part of the analyst, and the platform seeks to enable analysts to evaluate individual sources and data points in the course of the investigation.

Along with providing information for analysts to assess the quality of data points in an investigation, the object model combines DSRs and tethered sourcing with access control lists to support a fine-grained, multilevel security approach. As a platform designed for organizations that may work with data of varying levels of sensitivity (e.g. intelligence agencies), access control capabilities are a prime concern for Palantir. Consequently, the system is designed such that "each Palantir user can be assigned a series of access permissions that will enable the selective revelation of information based upon the user's particular role, mission, or authority, as well as his or her security clearance (where applicable)."[91] The platform produces censored views of PTObjects based on a user's access rights to the data sources that "support" the properties and relationships ascribed to ("contained" within) that object.

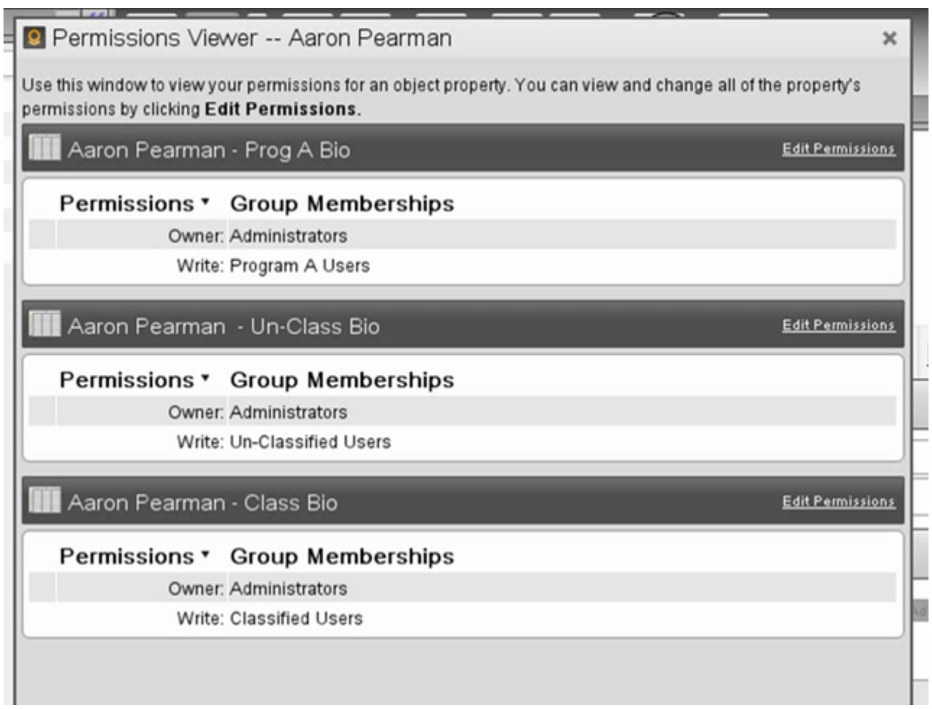

Figure 2.12: View of user permissions for data sources and derived properties in Palantir environment. Author screenshot from Palantir YouTube video, "Multi Level Security – Part 2."

Palantir implements what they describe as "sub-cell level security," referring to the cells of a table in a relational database and their platform's inclusion (and control) of a number of other data alongside the contents of that particular object property. Their training video on multilevel security describes each PTObject property as a "stack of cards," each of which has its own

[90] Palantir Technologies, "Object Model," YouTube video, 15:02, posted by "Palantir," July 5, 2012, https://www.youtube.com/watch?v=qNxk-1_r7A4.
[91] Palantir Technologies, "Protecting Privacy and Civil Liberties," 2.

access control settings.[92] An object representing a person "John Smith," for example, might contain properties for name, age, and address. Each of these properties in, turn, would contain 'cards' for: the value of that property instance ("John Smith"; "25"; "101 South Main Street"); a timestamp for when the data point was created; which user created the data point; the source or sources the data point is drawn from (including if it has been manually created); and the access control information for each of these cards. Access can be restricted to any or all of these components of the property – a user may have permission to see John Smith's age, but may be blocked from seeing where this data was sourced or which other users have access to this information. The platform supports five different degrees of access with respect to these cards. The first four degrees are intuitive for many computer users: ownership, read access, write access, or no access (which disallows knowledge of the existence of the data point in the system). A fifth degree, discovery access, may also be granted to users, in which a user is allowed knowledge of the existence of a data object (but not its content) and provided with instructions for gaining access to it – by sending a message to an administrator, for instance. The final user view of an object and its properties is calculated dynamically based on that user's permissions and the access controls for both the contents and metadata of those properties. Palantir claims that "access control adjustments take effect within seconds, instantly barring any newly unauthorized user from any further interaction with the newly restricted data."[93] The emphasis of this access model is not just stronger security, but more thorough collaboration, as the model "allows users with widely varying data permissions to work simultaneously with the same records."[94] In this, we see how deeper machine processes contribute to the collaboration across organizations and facilitate, at scale, the kind of analytical "choreography" featured in Chapter One.

Object model

A flexible or dynamic ontology should not be confused with an ontology that is completely blank. Though the Palantir object model is highly general, it does include a few "hard-coded" notions that constitute its default orientation toward the semantic of the datastore. New object type definitions are derived from three 'base types': entities, events, and documents. Many of Palantir's paradigmatic sources for 'character data' are transactional – network traffic and access logs, call data records, bank transfers, e-mails, and the like[95] – the result of an entity encountering some transducing structure and producing a flurry of persisted digital activity. A user login on a shared corporate network generates an entry in an access log. A phone call produces a call data record (CDR) on a telecom company's server. The swipe of an employee access card is recorded. Collections of character data are parsed and packaged into PTObject containers, with the end result that digital traces are extracted and re-presented as *entities* and *events* that contain or possess properties and relationships. *Documents*, which serve as containers for unstructured data, may also be linked to these objects through tags (which may also contribute data on properties). Intended to be general enough to seed the modeling of a wide

---

[92] Palantir Technologies, "Multi Level Security – Part 1," YouTube video, 5:06, posted by "Palantir," July 5, 2012, https://www.youtube.com/watch?v=rmppUPvc-tk.
[93] Palantir Technologies, "Protecting Privacy and Civil Liberties," 2.
[94] Palantir Technologies, "Multi Level Security – Part 2," YouTube video, 11:54, posted by "Palantir," July 5, 2012, https://www.youtube.com/watch?v=wNHfO4Xa9K4.
[95] Stowe et al., "System and method," Section 2.1

range of application domains, these three types (entities, events, and documents) serve as a kind of scaffold on which an organization can ‘hang’ more expressive terms for their data objects.[96]

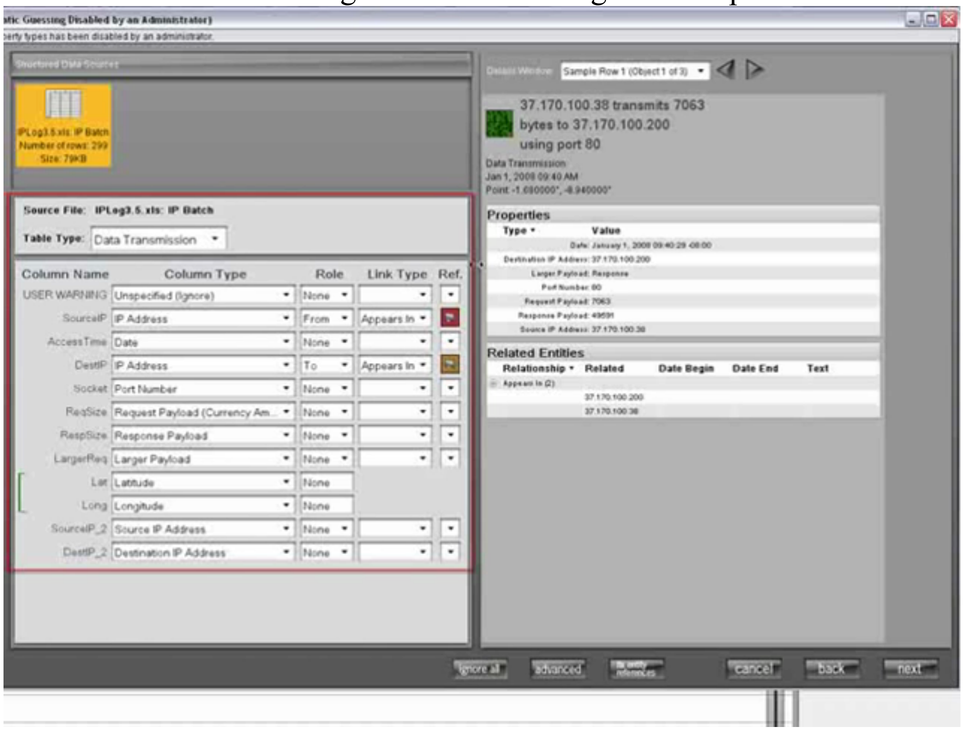

Figure 2.13: Importing access log data into the Palantir environment for representation as graph objects. Author screenshot of Palantir YouTube video, “Finding a Mole Cyber Counter Intelligence – Introduction.”

Despite what appears to be an unremarkable generality, we should not be too quick naturalize these base types. While the key-value driver and adaptor components enable the movement from “raw” formats to indexed, searchable key-value pairs, the ontology layer abstracts from these key-value pairs to present an analyst with “persons” linked by events like wire transfers and phone calls. Figure 13 shows one such instance with conversion of a data transmission log into a set of Palantir objects. These concepts (entity, event, relationship) and their extensions procedurally embed specific claims regarding the categories for effective reasoning in organizations, claims about how human beings “really think about the world”[97] – another set of techno-epistemological decisions that the platform enacts. The Gotham environment, for instance, logically and visually flattens all manner of relationships and interactions – kinship, party membership, phone calls, wire transfers – rendering them as lines (edges) on the object graph, a graph that could represent any number of domains without regard for their characteristic channels.[98] The graphs appear on a blank, white background, a network of entities presented in an eternal present. And while animated “flows” gesture toward the quality and temporality of a narrow range of relationships (monetary relationships especially), the graph hides most social and historical depth from the analyst – there is little room for contingency or situatedness in this representation. These hard-coded notions structure the semantics of the Palantir possibility space and invite human analysts to engage in a specific hermeneutic of social

[96] Palantir Technologies, “Dynamic Ontology,” YouTube video, 10:30, posted by “Palantir,” July 5, 2012, https://www.youtube.com/watch?v=ts0JV4B36Xw.
[97] Palantir Technologies, “Building a Human Trafficking Case from a Lead to Arrests,” YouTube video, 20:20, posted by “Palantir,” July 15, 2013, https://www.youtube.com/watch?v=gS1IMB--3dw.
[98] Palantir Technologies, “Object Model.”

action: Gotham investigations proceed via readings of digital traces as behavioral events, events attributable to definite entities with particular properties and relationships. The object model, and the more detailed ontologies that inherit the model, package data into these ontological units for later assembly into higher order explanations and interpretations that are attributed to the reality of the domain rather than the structure of the object model.

The progress of a short counterintelligence demonstration from Palantir provides a clear example of this hermeneutic as the analyst attempts to uncover the identity and social network of a "malicious insider" (a common theme in their demonstrations) who may be sending classified data from embassy computer systems.[99] The goal is to identify any suspicious data transmissions and attribute them to a probable suspect. Within the interface, an import wizard converts logs of network traffic, security badge events, and employee records into Palantir objects compatible with the Graph application, and these objects possess properties and relationships in accordance to the chosen ontology. Beginning with a visualization of badge event objects representing entries into and exits from the classified space in the embassy, the analyst highlights abnormal "open-ended" events, those without both an entry and exit time as properties, where an employee may have followed behind a colleague without swiping their badge. Noting that employee 30 "has an interesting weekly pattern" of this piggy-backing behavior, the analyst turns to the employee records to investigate which employee and computer entities share an office with the employee already linked to these suspicious events. Using the timeline, the analyst scans for data transmission events when the authorized user was outside of the classified space at the time. The analyst notices a transmission from employee 31's computer (employee 30's officemate) to an IP address outside the embassy. Adding this IP address object to the graph, the analyst performs a "search around" which returns twelve related embassy computers and eighteen linked data transmission events; the properties of these transmission event objects, summarized in the histogram, indicate transmissions of substantial amounts of data to the IP address in question over port 8080.

[99] Palantir Technologies, "Finding a Mole: Cyber Counter Intelligence – Introduction," YouTube video, 1:43, posted by "Palantir," July 5, 2012, https://www.youtube.com/watch?v=NanaHD7iMLQ. (As well as three subsequent videos, "Finding a Mole: Cyber Counter Intelligence – Part I," 3:53, https://www.youtube.com/watch?v=53NVooaR4mk; "Part 2," 4:02, https://www.youtube.com/watch?v=nzX1qRf80yc; "Part 3," 2:53, https://www.youtube.com/watch?v=Vn1a-hoBSn0.

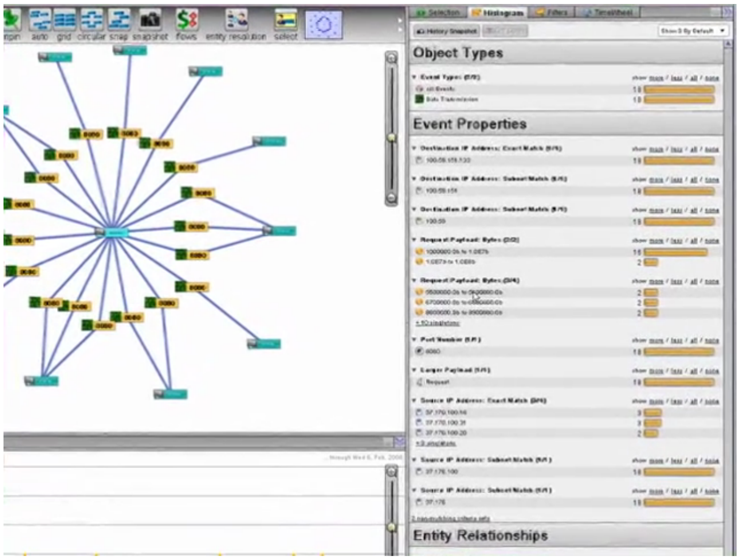

Figure 2.14: Graph and histogram representations of access log data in the Palantir environment. Author screenshot from: "Finding a Mole: Cyber Counter Intelligence – Part I."

Further analysis of the overlapping timing of these transmission event objects, the authorized users linked to those computers, and their badge events confirms the suspicion that employee 30 may be involved in the exfiltration of classified data.

The entities (employees and computers) and events (badge swipes and data transmissions) contain properties and links that suggest capacities and motives. The analytic products of these demonstrations, the final explanations of the data, emerge from cycles of suspicion, attribution, and relation. Each analytical step is driven by the hypothesis (or hope) that the next linked object will somehow "contain," in the form of its properties or other relationships, an explanation or the seed of an explanation for the events in question. The analytic aspiration is that those digital traces can now be seen as the effects of behaviors by named entities. The investigative field, as presented by the platform, is populated with actors with varying levels of dangerousness or risk. The first chapter ended by highlighting the role of narrative in Palantir Gotham investigations and their explanation of networked phenomena. In this light, we might see the 'cut out' objects produced by an ontology as providing source material for subsequent narrative sutures. Responding to Lev Manovich's assertion of the incompatibility of database and narrative forms, N. Katherine Hayles argues that these two technologies are actually symbiotic, with the temporal technology of narrative – oriented toward the inscription of agents with agendas and desires, motives and actions – selecting elements from contemporaneous space of the database.[100] While agreeing with Manovich's distinction between the field of paradigmatic possibilities present in the database and the syntagmatic choices that characterize narrative, Hayles contends that "narratives remain the necessary others to database's ontology," moving users from classification and enumeration to interpretation and meaning. In the language of programming, ontologies serve as semantic "hooks" (similar to API hooks) for data analysis and interpretation, supporting the assembly of a narrative with Palantir objects serving as agential pivots for sequences of events.

This computation and presentation of objects, properties, and relationships enacts a set of "agential cuts" encoded in the deployment's ontology. Karen Barad argues that objects (and their

[100] N. Katherine Hayles, "Narrative and Database: Natural Symbionts," *PMLA* 122 no. 5 (Oct. 2007), 1607.

representations) are not fundamental epistemological units, rather, they are temporary separations and stabilizations of the "intra-activity" of indeterminate phenomena.[101] The parser definitions in the ontology configure the system to carve out data entities from digital sources of various types, a material and discursive configuration in which a "'part' of the world becomes determinately bounded and propertied in its emergent intelligibility to another 'part' of the world." In Barad's view, the broad Netwonian ontology of independent objects, in which I would include the sociotechnical physics implied by the Palantir object model, is a performance, the result of exclusionary acts occurring within a mass of biological, electrochemical, and other phenomena. To discount performance and naturalize these categories is to fall into an epistemological hazard accompanying the practical use of computational systems that Brian Cantwell Smith calls an "inscription error." According to Smith, the inscription error is:

> a tendency for a theorist or observer, first, to write or project or impose or inscribe a set of ontological assumptions onto a computational system (onto the system itself, onto the task domain, onto the relation between the two, and so forth), and then, second to read those assumptions or their consequences back off the system, *as if that constituted an independent empirical discovery or theoretical result*.[102]

A critical view of the Palantir object model (and the ontologies it grounds) leads to an understanding of the platform as a set of sites or instances for this kind of performance in which a local, contingent materialization presents a view of independent objects and agents with attributable properties. The implicit design objective for the object model and dynamic ontology, with respect to techno-epistemology, is a machine whose logico-symbolic activity can quickly and readily contribute to the semantic strata of investigation and analysis, an objective partially achieved through the 'packaging' of digital traces as entities and events. This packaging, however, conveys a set of ontological commitments that may not be recognized as such, and narratives generated and meaning made with these commitments should be understood as contingent, chosen, and produced.

Meaning in the machine

Considering Palantir deployments in organizational contexts as composite (H-LAM/T) systems, tasked with solving the intellectual problem of anomaly detection, reveals a further implication for the 'packaging' of traces as conceptual PTObjects according to a dynamic ontology. The ontology and its constituent concepts – of various entities, events, and relationships – serve as a "medium of exchange," transmitting preferred modes of comprehension to analysts across the organization, constraining their investigative manipulations and views of the data and guiding their analytic movements.[103] Roughly corresponding to what Engelbart calls the concept structure of the artifact, the ontology provides a common language and a consistent set of intellectual priorities for analysts engaged with the platform and its data. Engelbart describes computing artifacts as elements of a composite system that encompasses both man and machine, and the computer can do more than accelerate intellectual activity – it can change the character of those intellectual processes. In this respect, Engelbart's model

[101] Karen Barad, "Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter," *Signs* 28 no. 3 (Spring 2003), 801-831.
[102] Brian Cantwell Smith, *On the Origin of Objects* (Cambridge, MA: MIT Press, 1996), 50.
[103] Engelbart, *Augmenting Human Intellect*, II.C.5.b.3).

intersects with the work of Ernst in particular (and other critics of media, technology, and society more generally) in drawing an artifact's technical make-up closer to its intellectual, social, and epistemological effects. A common language of concepts structures the analysis products that emerge at the man-artifact interface, the meaningful narratives posited by the human analyst – especially those narratives that surround exceptions, threats, and interruptions. In this respect, ontology moves us away from the 'deeper' explicit-artifact processes – those in the datastore that address the storage and retrieval of sets of data objects as arbitrary groups of symbols – toward the man-artifact "surface."

Palantir's dynamic ontology recalls an important distinction Engelbart makes between two related classes of intellectual activity: the direct-contributive class, composed of those tasks that are closely linked to the realization of a given product; and the executive class, composed of the planning and policy making, that is, the process development tasks that create guidelines and structure the direct-contributive efforts. We might understand the dynamic ontology feature in Palantir opens up a deployment's concept structure to scrutiny and modification and attempts to accelerate the implementation of the results of metacognitive work by introducing a new layer of abstraction. The introduction of this top-most interpretation layer makes this possibility for reprogramming concept structures a more integral part of the platform, a less exceptional possibility for the computational artifact. If, as Barad claims, such conceptual structures and the various "cuts" they make with respect to the phenomenal world are the products of material configurations – the rapid electrochemical pulses of crystalline silicon – with a weight and inertia of their own, the dynamic ontology feature can be seen as a way of keeping this material-conceptual apparatus "up to speed" with the domain logic after deployment. Palantir claims that, through the dynamic ontology feature, the conceptual structure of the data platform "can grow along with your organization's view of the world."[104]

But this reprogramming of concept structures, as discussed above, is not solely focused on an abstracted logic of the domain; it must also consider the ultimate task of producing narratives with these data objects. Orit Halpern locates ambitions for the productive entanglement of computer-based visualization, knowledge, and narrative structures in the post-World War II work of designers like Gyorgy Kepes and Charles and Ray Eames and their efforts to develop a new form of observer. Halpern's reading of Kepes's work highlights his pursuit of "a 'new structure-order' […] that emerged from the recombination of vast data fields."[105] The new design, for Kepes, could provide an engine for inquiry and knowledge through its organization of heterogeneous media—text and visual images—according to a "common structure," prompting this observer to make stories and make meaning as product of creating connections between these data objects.[106] Kepes's various experiments explored this potential for designers (and observers in general) to "agglomerate information and retroactively discover patterns" in a relational process that was "equated with perception and cognition simultaneously."[107] The Eameses' experimental course in cybernetics and communication theory exposed students "to a vast amount of data" with the goal of having students "unearth a distilled logic or pattern within a non-media-specific data field constituted of sound, smell, and image."[108] If presented in accordance with the proper structural principles to this new kind of observer, a

---

[104] Palantir Technologies, "Dynamic Ontology."
[105] Orit Halpern, *Beautiful Data: A History of Vision and Reason since 1945* (Duke University Press, 2014), 87
[106] Ibid.
[107] Ibid., 95.
[108] Ibid., 105.

collection of objects, diverse in kind and content, could be reduced to their essence in a process that was both affective and logical.[109] The consequences of structure, of a datastore ontology, lie not only with their representational qualities but also with their inspirational character. The concept structure described by Engelbart, then, may be seen as a medium of exchange for both information *and* affect.

Modern organizations are often conceived of as informatic organisms, decision-generating assemblages constituted by rationality, information, and communication. By loosening the coupling between the concept structure and the storage logic of the platform – between the ontology and the keyspace – an organization can more quickly align the temporality of their information and analysis infrastructure (with its complex microprocessual timings) with changes in their mission, scope, and business logic. With respect to the mnemonic logic of the platform, the object model and ontology offer up a semantic surface, an organizational logic, for the production of narratives – narratives of infiltration, of fraud, of threats and vulnerabilities. In the language of actor network theory, database ontologies facilitate the concerted enrollment of human analysts and computerized data objects in such narrative production. These ontologies are materialized concept structures, aspects of the machinic process and its n-dimensional internal image. These structures constrain, communicate, specify, and stabilize semantic elements and narrative synthesis, reconfiguring circuits and temporarily reorienting digital micro-processes in the direction of specific patterns of affect and genres of narrative—that is, reorienting them toward culture.

7. Conclusion

Design and analysis of computing machinery (as Agre describes) materio-discursively organizes digital intra-activity into layers of abstraction of different degrees or heights. This chapter has moved through descriptions of three successive layers of abstraction in the Palantir platform: distributed datastore, key-value driver, and the dynamic ontology. Sets of differently structured processes are concerned primarily with specific aspects of the apparatus and task environment, and each set is implicated in the realization of a different, limited set of parameters for the system. The distributed datastore, discussed in sections two and three, sits closest to the processes of physical storage and data transfer among machines. Using Cassandra and Dynamo as exemplary (but not exhaustive) cases, I argue that the distributed key-value store's design is organized around the tension between fragmentation and coherence. The algorithmic synthesis of a keyspace and the computational (and energetic) overhead of data and metadata transfer continually ease this tension. Mechanisms for coordinating the storage and retrieval of data objects and maintaining their consistency mitigate the tendency of an overlay network to dissolve and get out of sync, a continuous performance that prevents the system from reverting to separate, uncoordinated autonomous machines. Though in the technical literature *anti-entropy* describes only one subset of approaches for updating replicated databases,[110] the concept aptly sums the challenge facing those designing distributed database systems.

The consistent hashing of the key-value store presents us with a flat, homogeneous keyspace, with the system doing not much more than linking identifiers to locations, and making

[109] Ibid., 133.
[110] Demers et al., "Epidemic Algorithms."

sure the data objects at those locations are up-to-date. As it is, this system does not account for any logical links between the data objects – one ready comparison would be a library that had no way of readily retrieving all the books penned by a single author. Here, the tension emerges between the relatively simple, homogeneous, and coherent keyspace produced by the distributed datastore – one dimension, as each object can be specified by a single coordinate designating its position along the circle – and the desire for access and operations on data objects along additional dimensions, representing their variety of types and multiplicity of attributes. The key-value driver (described in the fourth section) resolves this tension through a kind of logical overlay. The driver consists of procedures that embed and process indirect references in the keys and values from "above" the datastore, as it were. In one example, we saw how such a system might represent e-mail users and their sent messages as distinct (but linked and searchable) logical data entities. In this manner, the system as a whole augments the functionality of the distributed datastore which treats each key-value pair as an isolated data object.

With the procedures of the key-value driver, a client application has access to a multi-dimensional space of data objects, permitting the search and retrieval of items according to their different types (e.g. user or message) and attributes, as well as the ability to store and access objects as being linked to one another in some fashion. No semantics inhere in such a system, however. Training, explicit textual labels, or other features may succeed in teaching a new user of the system precisely what kinds of things are represented in the datastore, but such disambiguation is post hoc and is not integral to the operation of the datastore. Moreover, a change in the universe of discourse (the types of objects, attributes, and relationships represented) may require a reprogramming of both the key-value driver (key-value families and permissible links between them) and revision of any ancillary materials that elucidate the semantics of the key-value pairs and their relations. Here, a tension emerges between the material-conceptual inertia of a present implementation – the categories and relationships, the parsing and importing procedures already programmed that organize the database – and the requirements of future abstractions that will accompany inevitable changes in available data sources, business logic, and organizational scope. In turn, the dynamic ontology (section five) resolves this tension between a finite artifact and an uncertain future through an additional layer of interpretation (and indirection). PTObjects serve as "containers" for the data points stored in the key-value store, but these logical object containers remain distinct from the data points that constitute them. These object containers serve as an initial channel for requests by client applications for object's associated data points. With this approach, the semantics of these containers can be modified without the reprogramming disrupting the constitutive key-value pairs in the datastore. New categorical and, consequently, narrative understandings of a domain can be rapidly integrated into the organization's analytical infrastructure.

Throughout this chapter, I have highlighted the tensions between abstraction (the logical structure and definitions that make up the functional aspirations of the platform), and implementation (the feasible elements and configurations for realizing such structures). These technical elaborations set out the governing principles of a material-conceptual apparatus that enables and constrains the production of statements and narratives of anomaly from database objects cut out from heterogeneous streams of information – a digitally articulated archive, in the Foucauldian sense. Further framing this discursive production are the networks that Palantir investigations aim to elaborate and comprehend. These networks prominently involve the movement of information and/or material through high volume systems – currency transfers, phone calls, e-mails, maritime container traffic. While sociotechnical systems serve as a ground for widening circuits institutional surveillance and control, they also present opportunities for actors to evade, thwart, and undermine the agential identifications that facilitate administration. Actors can hide in the noise. Large scale digital infrastructure provides for a wider visibility and scope of normalization, but this visibility and scope emerge alongside the expanded possibilities to mime "normal" behaviors while furthering contrary objectives. Moreover, scale is not the only issue; the multiplicity and heterogeneity of these systems is a prime concern for the Palantir platform. Activities may be segmented across disparate systems whose monitoring may not be coordinated. The multimodal action of dangerous networks is a particular design *agon* for Palantir, and their emphasis on human-driven investigation can be read as an address to this heterogeneity. The Palantir environment encompasses a broader contemporary tension between the capacity of digital techniques to provide a surface for visualizing the social *and* their capacity to segment and obscure actions and their origins.

The challenges that Palantir cites for organizations, those of scale and heterogeneity, center on this tension, the potential for computing machinery to alternately thwart and enable the work of institutions in positing a narrative, agential arc accounting for a set of digital traces. As a platform, Palantir indexes a prime concern for institutions with reconstructing and stabilizing agential arcs, with the potential for narrative, traceability, and accountability. The scale and heterogeneity of contemporary digital infrastructures can frustrate efforts to catalog and comprehend sociotechnical activity in agential terms, to group and ascribe digitally mediated events to a conventional subject – we might say, to a desiring consumer vulnerable to the economic influences of the market and a bodily citizen vulnerable to the coercion of the state. Palantir indexes a concern with maintaining the capacity for creating signatures, for individualization, and for corrective (including coercive) intervention – for updating the secondary (but still active) enclosing and repressive functions of institutions in a dispersive, undulatory, digitally-inflected world. The platform, as an exponent of "organizational intelligence," facilitates the "layering" investigations and the creation, from streams of digital traces, of a "thicker," narrative world of objects, agents, motives, and relationships. Palantir makes mediated actions and "proper" identities explicit such that bodies can be accounted for by institutions, in relation to and as a function of digital traces, and to counteract the tendency of subjects to recede into the undifferentiated intra-activity of digital phenomena.

Chapter 3
From Threats to Opportunities: Seeing Connections and Making Markets in Metropolis

1. Introduction

Digital systems enable and track activity in almost every imaginable domain, generating vast numbers of records or traces every second – in 2010, Google CEO Eric Schmidt famously (perhaps with a touch of hyperbole) claimed that the world was creating five exabytes of data every two days.[1] A claim commonly accompanying talk of the "data deluge," especially from makers of systems like Palantir, is that these data alone are not a source of competitive advantage or value. Analysis is necessary to transform data into information, to move from a collection of records to claims about trends and anomalies in the sociotechnical systems that generate those data. This information, extracted and synthesized from "raw" data, is the stuff of value and significance that improves decision quality for organizations. However, meaningful, relevant connections within datasets may be elusive and, as seen in the Gotham demonstration, require multiple cycles of speculative search, evaluation, and revision. Meaning depends on the deployment of systems for working on data, for their analysis and interpretation. This subordinate role of data is neatly expressed by a Palantir speaker claiming that their company are about "big information, not big data."[2] This hermeneutical presumption, that meaning is not immediately evident but only becomes so through the work of interpretative systems, is a critical part of the value proposition for Palantir's products and others like it.

Chapter 1 examined Palantir Gotham and found that the sought-after connections, anomalous and elusive moments, were often instances of unexpected and/or illicit structures of organization among objects. The analysis proceeded via human engagement with the quality of these structures – adding, removing, and rearranging nodes to draw out and emphasize those qualities – through their visual representation in the Graph application. In Gotham, these anomalous community structures, appearing in security domains like law enforcement, network intrusion, and disaster response, were pursued not only as signs of potential threats to the organization but also, in their mapping of influential and critical relationships, rubrics for intervention or what is euphemistically termed "proactive community management."[3] Investigative intelligence appeared as a synergetic product between human analyst and computer platform, a digitally assisted endeavor of assembling traces of actions cut out from heterogeneous datasets into hypothetical narratives invoking style and motive primarily mediated by the Graph application. The coupling of this investigative environment with Palantir's distributed datastore and dynamic ontology constitutes a platform that coordinates the timing of human experts with that of digital systems and traces, the target domain, and the organization as a whole. In particular, this use of the graph or network as an image-concept in Gotham may embed an epistemological ambition to increase the amount of relational data available to analysts, an ambition directed toward the theoretical possession of a near total graph of the

---

[1] Josh Halliday, "Guardian Activate 2010: live coverage," *The Guardian*, July 1, 2010, https://www.theguardian.com/media/pda/2010/jun/30/guardian-activate-summit-2010-liveblog.
[2] Palantir Technologies, "Humans, Data, and the Culture of Too Much information" (see Introduction n. 8).
[3] Benedikt Boecking, Margeret Hall, and Jeff Schneider, "Event Prediction With Learning Algorithms—A Study of Events Surrounding the Egyptian Revolution of 2011 on the Basis of Micro Blog Data," *Policy & Internet* 7, no. 2 (2015): 159-184.

domain to certify investigative results. The graph as epistemic mediator also embeds a tension between the network as an epistemological device and as a social fact, and this network thinking plays a significant role in the operations of organizations using tools like Palantir Gotham to inhabit and administer their data-based realities, including prompts and justifications for decisions to investigate, surveil, arrest, and even to kill people "based on metadata."[4] The systems of interpretation that extract knowledge are entangled with the patterns of ensuing actions.

This chapter takes up Metropolis (previously Palantir Finance), an alternate version of Palantir's platform oriented toward quantitative data with suggested applications including "insurance claims data, network traffic flow, and financial trading patterns."[5] As with Gotham, Palantir emphasizes the Metropolis platform's capacity for wide-ranging data integration, intuitive data modeling, and ease of extensibility within a single analysis environment. But while Gotham is primarily constructed around the investigation of anomalies as threatening instances with the potential to disrupt the positive work of an organization, Metropolis is centered on the recognition of anomalies as opportunities to be exploited for future financial gain. Gotham sought to support the analyst in the discovery of illicit structures hidden in the vast amounts of data that, to a great degree, reflect normal behavior, elusive community structures. Metropolis, by contrast, approaches the elusiveness of meaning in capital markets. Put broadly, two major theoretical innovations in finance, random walks and market efficiency, seem to preclude the possibility of using price movements alone to justify any trading strategy. Of course, profits in capital markets are made and often on a systematic basis. This chapter is about how Metropolis enables certain patterns of interpretation for market data in order to reveal meaningful connections therein, connections that can be exploited for potential profit.

Metropolis is an extension of the Palantir experiment and design philosophy to quantitative data, financial data especially, and the opacity of capital markets. The Metropolis platform iterates the commensurability of social domains suggested by Gotham in a direction, shifting the data model but ultimately making a similar statement about the analytical yield of composite human-machine systems, where this composition is facilitated by intuitive object models, flexible ontologies, interoperable visualization interfaces within a unified environment and supported by a fast, distributed datastore. Commensurability, however, is not a feature given in the nature of any domain. Rather, it is a construction of the epistemic apparatuses that make knowledge about those domains; commensurability is a performance, one often inspired by extra-theoretical concerns (e.g. cost). The ways in which organizations attempt to make domains commensurable may be clues to the techno-political aspirations. These are not aspirations to a knowledge of disembodied facts but rather actionable information – after all, intelligence is about adaptation and decision, its activity attempts to remake the world according to certain goals.

This question of the effects of knowledge practices has emerged in the literature surrounding economic rationality, markets, and interpretive devices, one that both by analogy and by its entanglement with the politics presents important frameworks for understanding the consequences of analysis platforms and infrastructures. Specifically, Metropolis evokes interpretative market devices whose incorporation does not entail the realization of some discrete proposition about prices or other quantitative indicators within financial markets. Rather,

4 Cole, "'We Kill People Based on Metadata'" (see chap. 1 n. 88)

5 "Palantir Metropolis | Palantir," Palantir Technologies, accessed October 3, 2016, https://www.palantir.com/palantir-metropolis/

Metropolis and these devices (models, interfaces, chatter) call our attention to the ways that devices modify markets as a kind of possibility space, in particular changing the things it is possible to see together and see as connected, enabling different kinds of action in response to that vision. Metropolis, I will argue, puts forward a grammar of analysis, and grammars of analysis become part of patterns of action.

2. Joyride: Touring Perspectives, Looking for Possibilities

Palantir's demonstrations for both Gotham and Metropolis frequently begin with and are punctuated by phrases like "one might ask the question…" or "perhaps you'd like to know…", highlighting the design of their products as responsive digital artifacts. When distinguishing both Palantir platforms from other analysis tools and describing their unique value added, a related refrain from Palantir speakers is that the platform enables users to focus on *asking questions*. The user does not spend valuable time (and expertise) translating the question into a form compatible with the database, spreadsheet, or other legacy software system, Palantir claims that its interface (and underlying technologies) provide more intuitive ways to interact with data that allow non-technical users to ask sophisticated questions. Palantir's interface and data model reflect "the way the world really works," and are compatible with the way analysts really think, not in terms rows and columns, but in terms of objects.[6]

In the Metropolis platform, Palantir presents financial analysis as a set of logical building blocks, a hierarchy of financial objects whereby the analyst, in generating and manipulating these objects in a *visual* environment, can quickly and intuitively explore hypotheses and answer questions using the quantitative data packaged in those objects. The application tabs at the top of the screen, the different "rooms" for analysis, are each oriented toward a specific mode of interaction with the basic objects of the platform.

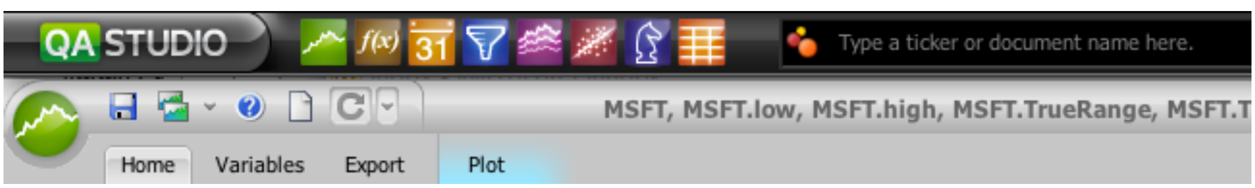

Figure 3.1: Application bar for Thomson Reuters QA Studio, an alternate version of Palantir Metropolis. From left to right, square icons indicate the following applications: Chart, Custom Metric, Date Set, Instrument Explorer, Index, Regression, Strategy, and Spreadsheet. From Palantir Blog post "Turtle Breakout Trading Strategy Simulation."

With the Explorer application (or Instrument Explorer), users can search and filter financial instruments (stocks, bonds, currencies) based on chosen criteria and then save those instruments as a group for quick use in later analyses. Similarly, the Date Set application lets the analyst search for time intervals where certain kinds of market conditions obtained. The Index application allows users to create and save custom, aggregate measures to track overall performance for groups of instruments. In the Chart application, the user can create visualizations based on the characteristics of an instrument, a group of instruments, an index, or most any other object. Moreover, the user can drag and drop objects between the different applications or, more commonly, create their own custom names for user-generated objects

[6] Palantir Technologies, "Building a Human Trafficking Case from a Lead to Arrests" (see chap. 1 n. 2).

which are then indexed in the datastore and can be typed into relevant fields with the help of autocomplete.

Palantir explicitly presents Metropolis as a platform for what they see as a new kind of "object-oriented analysis" with respect to financial markets, providing users with interface objects reflecting the "building blocks" of financial analysis. Adjacent to this discursive claim are what Ian Bogost would call the procedural claims of the software itself regarding the relevant epistemic performances and the nature of the domain itself. Made through the representations, attitudes, and possibilities offered to users in their analysis and interaction with the financial market data, the procedural claims of the Metropolis platform emerge from its concrete digital operativity. Through the viewpoints, trades, and transactions that these procedures justify, the platform becomes entangled in the network of effects that constitute markets. Before situating Metropolis within a larger scheme of actors, devices, and infrastructure that make up financial markets, I will examine some prototypical patterns of interaction, or workflows, and begin describing, at the level of user interface and investigative logic, the orientations of the platform as a possibility space.

"JoyRide," the lighthearted title for an open online demonstration version of Palantir's finance platform, captures one such orientation. Available online to prospective clients from 2009 to 2013, Joyride showcased QA Studio, a version of Palantir Metropolis (then Finance) developed in partnership with the media firm Thomson Reuters and using content from Thomson Reuters's QA Direct product.[7] In particular, the tongue-in-cheek name presages important features of the platform in its suggestion of unstructured movement through space, an openness to new experiences and perspectives, and a sense of machine-accelerated opportunism.

A presentation at the 2011 event "Palantir Night Live: At The Highline," entitled "Palantir at a Global Macro Hedge Fund," features two employees from the Connecticut firm Bridgewater Associates, their head of research and a senior engineer, discussing the ways in which they have deployed and adapted the Palantir Metropolis (then Finance) platform to improve their research and strategy development processes.[8] As a global macro firm, Bridgewater aims to have "the deepest possible understanding of the global economy and financial markets," one that reflects a "fundamental, cause-and-effect understanding of markets," and to translate "that understanding into great portfolios and strategic partnerships with institutional clients."[9] With the caveat that they do not want to reveal too much of the Bridgewater "secret sauce," their firm's unique and systematic view of global markets, the two presenters guide the audience through some of the ways in which they use Metropolis to produce and systematize insight into global markets. The Bridgewater research director's initial overview includes evaluations that reflect much of Palantir's own design language surrounding the products, including the utility of a unified environment, the ease of collaboration therein, and their drive to reduce "friction" in analysis processes. The software demonstration that follows is staged as a conversation between the researcher and the engineer, with the researcher prompting the engineer for guidance on how to pursue lines of inquiry typical of the investment approaches at Bridgewater. The conversation and demonstration are intended to loosely follow

[7] "Welcome – Palantir JoyRide," Palantir Technologies, https://joyride.pfinance.com/welcome. Page removed, archived version: http://web.archive.org/web/20120504080123/https://joyride.pfinance.com/welcome/.
[8] Palantir Technologies, "Palantir at a Global Macro Hedge Fund," YouTube video, posted by "Palantir," July 5, 2012, link
[9] Bridgewater Associates, accessed August 6, 2017, https://www.bridgewater.com/.

Bridgewater's three-phase business process, composed of understanding, systematization, and productionalization. The research head emphasizes that an important part of their rationale for choosing the Palantir platform was its ease of integration with this conceptualization of the practices of financial analysis and investing.

Beginning with their first phase, understanding, the Bridgewater researcher opens the demonstration by describing for the engineer what he needs for a typical morning entering the office: a bird's-eye view of the current economic and market indicators, a high-level perspective on the day's financial landscape. Metropolis comes with a built-in Dashboard application which "allows users to visualize collections of analysis on a single screen," mixing visual and textual data in a customizable layout that can update in real time and promising analysts and organizations an interface reflective of their institutional priorities and understandings.[10] The Dashboard application "acts as both a place to consume the output of complex analysis and a creative springboard for new lines of inquiry," augmenting the situational of the researcher.[11] In this, the Dashboard can represent a kind of capstone for the firm's current analytical orientation and unique views of markets.

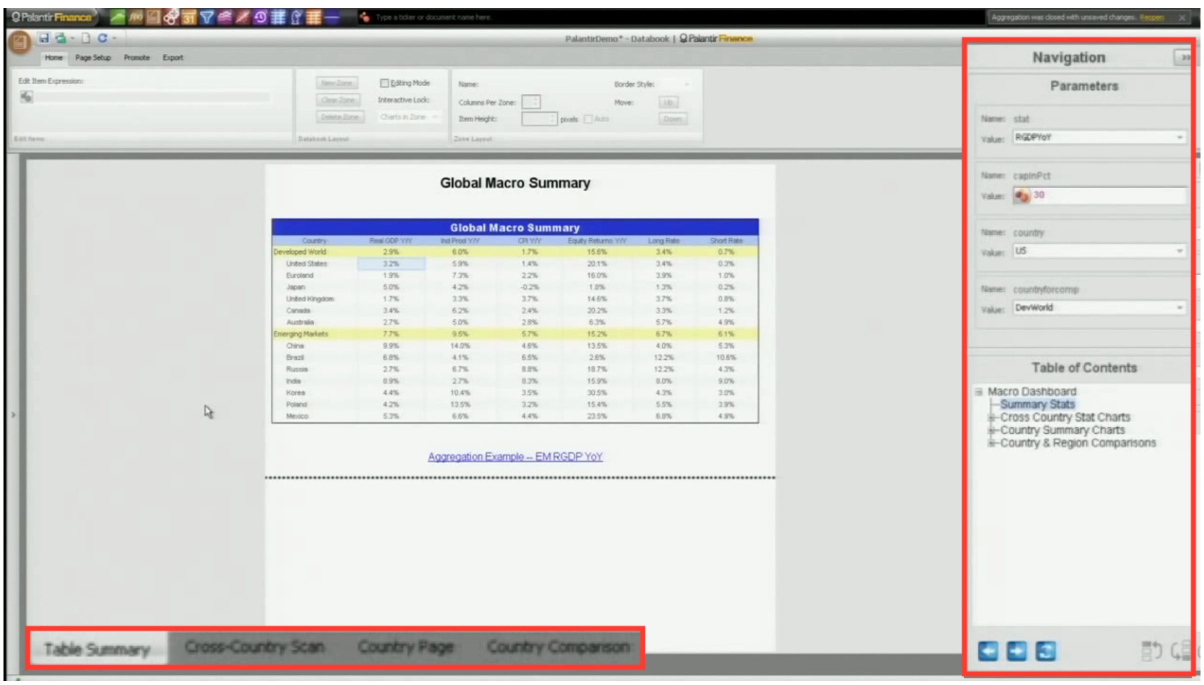

Figure 3.2: View of Bridgewater's Databook application, a replacement for Palantir's built-in Dashboard. Near the bottom are tabs for "Table Summary", "Cross-Country Scan", "Country Page", and "Country Comparison." The right-hand Navigation sidebar includes a dropdown menu for selecting display parameters, and a "Table of Contents" reflecting the common viewing order for the tables and charts. Author screenshot from video "Palantir at a Global Macro Hedge Fund"

Citing the firm's "specialized… fairly sophisticated needs around how to do visualization and data exploration and presentation," the Bridgewater engineer brings up Bridgewater's customized version the Dashboard application called "Databook." The engineer does not point to any specific deficiency in the default Dashboard application, rather taking the opportunity to tout the extensibility of Palantir in enabling them to easily produce and integrate their custom

---

[10] "Palantir Metropolis Applications | Palantir," Palantir Technologies, https://www.palantir.com/palantir-metropolis/applications/

[11] Ibid.

solution. Both Palantir's Dashboard and Bridgewater's custom Databook share an emphasis on seeding the user's understanding of current, chosen market conditions in a visual and, moreover, an interactive manner. The Bridgewater engineer highlights the multiple tabs in their Databook application which allow for different views of the data, and the ability to "navigate across those things in a bit of a narrative or story way." Taken together, these statements suggest an interface that, in the first instance, is oriented toward users' narrative sensibilities as a prelude to more fine-grained analysis.

Rather than a static report, Databook and Dashboard more closely resemble a staff briefing, where the user can request details or alternative views of the data initially presented. We see this when the head of research asks how, based on what he has seen in the Databook dashboard, he might use Metropolis to "drill down" on the United States and start to engage with the data underlying these aggregate measures. The engineer responds with a click on the Country Page tab near the bottom of the application, and selects the United States from the Navigation sidebar, displaying a page of charts for year-to-year real GDP versus industrial production and others economic indicators.

Figure 3.3: Country page for United States in Bridgewater's Databook application within Palantir Metropolis. Author screenshot from "Palantir at a Global Macro Hedge Fund."

Along with the detail provided by the Country Page tab in Databook, the engineer highlights how Metropolis allows the user to click and immediately "open" any number, including those presented in the Databook table overview, and "see" the time series behind it – specifically, opening a pop-up window with a chart displaying the changes in value over time from which the aggregate figure is derived. Using mouse gestures, the user can also zoom in, zoom out, or scroll to view different intervals of the time series data. The Bridgewater engineer explains that many interface components that might be static in other environments – numerical results, chart displays, and tables – become "rich" components in Palantir Metropolis, permitting a range of alternative interactions, for instance, flipping back and forth from a higher-level analytical product to explore its "supporting" objects, both data and methods. This part of the demonstration displays features of Metropolis centered on the user's movement between different, interlinked visual and logical perspectives on the data. The dynamic visualization of the time series that allows the analyst to put their trained eyes to work with respect to different time scales without interruption. Such interactions are made possible, in part, by Palantir's packaging of data and methods into discrete data objects. The displayed numerical results become "richer" when rendered as the dynamic outputs of these objects, ultimately providing an interactive path back to the underlying data and logic of their production.

The brief introduction to the Databook and Dashboard serves to demonstrate how Bridgewater uses Metropolis to support the "understanding" part of their research process, with the research head asking the audience to imagine how this might function at scale – more analysts, more countries, more data sources, and the ability to "navigate around and see lots of things." The lack of precision in this statement by the Bridgewater engineer belies a key feature of the platform, that of enabling, in the first place, lightly structured and open-ended surveys or navigation of the visual, numerical, and logical aspects of the organization's data and information assets. As the analyst first sits down in front of their terminal, the Dashboard application serves to update existing narrative understandings about their domain – national economies and financial markets, in this case – at a high-level while preparing for more specific and systematic analyses. The Metropolis platform provides an environment of optical and logical consistency where, as Latour says of early modern European techniques of inscription, data can undergo "translation without corruption" and become part of a synoptic perspective.[12] The synoptic perspective provided by Metropolis not only allows the analyst to speak (or think) "with many absent things presented all at once" but to do so with things presented at different scales.[13] The patterns of "thinking with eyes and hands" thus far suggest that the platform and its perspective should serve to renew analysts' broad awareness of domain narratives while keeping them open, opportunistic, and sensitive to different starting points for in-depth research.

The demonstration then turns from this lightly structured "understanding" phase of Bridgewater's research to "systematization"; the example of systematicity given here is the creation of a custom "propulsion index" aggregating the measures (industrial production and GDP among them) seen previously that would give the researcher "a sense of how much the economy is moving." In response, the engineer introduces Metropolis's Custom Metric application, where the user can write code in Palantir's Hedgehog programming language (or HHLang), an object-oriented language similar to Java, to express their ideas algorithmically, allowing the user or organization "build up a specialized analytic vocabulary."[14] Using this tool, the engineer writes a short program to combine the relevant time series and generate the index.

---

[12] Latour, "Visualization and Cognition," 8 (see chap. 1 n. 50).
[13] Ibid.
[14] "Palantir Metropolis Applications."

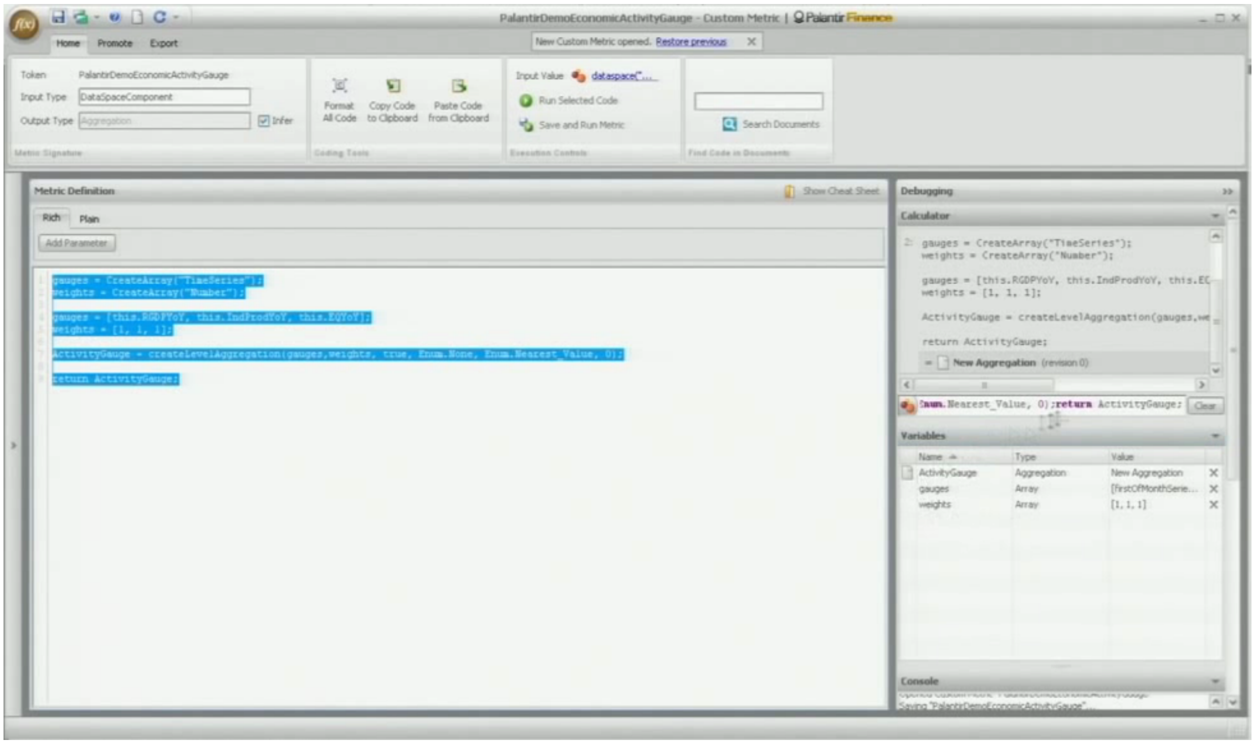

Figure 3.4: Creating a custom metric for economic activity in Metropolis. Author screenshot from "Palantir at a Global Macro Hedge Fund"

As a development environment in miniature, the Custom Metric application can run the HHLang code in a stepwise fashion, with a sidebar for displaying variables and values for the metric code and other information for debugging. In response to the researcher's request for a propulsion index, the engineer shows us a recently created "ActivityGauge" metric within the Custom Metric application which uses a function "createLevelAggregation" to combine the time series data for real GDP, industrial production, and an equity market measure for a selected country. Again, the engineer points to the ability of Palantir to link numbers to interactive or "rich" *documents* that, as in the case of the Databook dashboard, can be opened and explored in more detail.

The interface connection between displayed results and their supporting documents suggests an emphasis on continued openness to the adjustment and revision of analysis within Metropolis. Using another custom Bridgewater application built with the Palantir API called "Aggregator," the engineer gets an interactive display of the resulting time series adjacent to a color-coded chart of the contributing objects, as well as a panel for modifying the aggregation components and weights. The Aggregator application allows for the visual exploration of the results of the ActivityGauge metric as written, as well as the ability to tweak parameters displayed in the panel and immediately see those results, a useful outcome of Palantir's "snappy connection between logic and visualization." Looking at the chart of the aggregated time series, the researcher has an inkling that the weighting of one of the data series may be a bit too high, and he expresses a desire to revise the custom metric just created. After generating a new chart that looks more "reasonable" to the researcher, the engineer quickly changes the original custom metric code to reflect this revision. The custom ActivityGauge metric is then saved as a reusable object (a document in the language of Metropolis) in the firm's datastore. The systematic understandings encoded within Metropolis invite tweaks and revisions, and they can be quickly made a part of the analyst's, department's, or entire organization's tools of perspective.

In summarizing what Bridgewater employees have found "pretty cool" about the Palantir platform, the engineer names three features. First, there is the ability to view both "raw" and derived data in the same visual environment, as well as to modify the code within that same environment and see the resulting changes in those visualizations – an enthusiasm we saw in the brief description of the ActivityGauge metric. A second feature touted is the ability to "apply ideas across different countries and markets," that after the development of a new metric, for instance, this metric can take as input any compatible object within the firm's datastore and generate the desired analytical product. The third feature he praises is the inclusion various "knobs" adjacent to the display of results and visualizations, readily available interface elements to adjust the parameters for a calculative methodology in the metric code, which leads to dynamic analysis products that their researchers and investment officers can "manipulate and play with interactively," at least partially in the service of certain heuristic tests (visual or otherwise) that aim to establish the reasonableness of a line of inquiry. The value-added or the "power of out of the platform" put forward in the Bridgewater presentation centers on Metropolis's functionality as a cohesive but flexible possibility space, one where the logical and algorithmic links between visual presentations, numerical results, and underlying data objects can be generated, examined, and adjusted within the unified environment.

3. Time Series, Grammar, Laboratories, Linkages

The patterns of interaction in the Bridgewater demonstration tended to emphasize a heuristic survey of visualization, but near the end, in the creation of the propulsion index, we saw a gesture toward the creation and modification of the calculative logic behind the charts. A key methodological claim for this dissertation has been that the procedural characteristics, observed at the level of interface and user interaction, are emergent features. They appear at a high level of abstraction as the result of combined lower-level operations (down to the semi-conductor level). Making procedural readings a component of a larger reverse engineering process entails a further examination of these underlying levels of digital operativity in order to understand the ground of epistemic performances in Palantir. The models and assumptions implied by the observed workflows are only a part of this ground. Looking at things like the datastore and object model, not just the interface, as a relevant part of possibility space highlights the discrete components of these interactions and their potential for novel and unanticipated combinations. This potential is, for Palantir, what it means to build a *platform* and not some other kind of software product, the chance that "when you give it to someone and they build something that you didn't anticipate."[15]

Palantir takes the spreadsheet as an initial point of repulsion, claiming to have begun with "the legacy model of building a monolithic analysis in Excel… and transformed it into many individual building blocks," and their object model is based on the resulting "conceptual, logical building blocks."[16] These building blocks, in Palantir's view, "[create] abstraction that helps to control complexity" compared to spreadsheet-based investigation, facilitating collaboration and innovation as both results and elements of analytical methodology are stored as discrete, re-

---

[15] A comment by the opening Palantir speaker in the presentation "Palantir at a Global Macro Hedge Fund."
[16] Palantir Technologies, "The Palantir Metropolis Object Model," YouTube video, 15:45, posted by "Palantir," July 19, 2012, https://www.youtube.com/watch?v=HSIbUt06Hoo.

usable and re-combinable instances of object types within the datastore. The plane of consistency from the previous section is, in part, a visual overlay of interaction possibilities for the Metropolis object model. Examining the other levels of abstraction links the platform to wider debates in human-computer interaction and epistemology; more importantly, it enumerates the "building blocks" whose programming constitutes the platform and participate in its network of effects – epistemic and otherwise.

Palantir designed the Metropolis object model as an approach to intuitive (but also complex and flexible) interactions with the given or raw data of financial analysis, the records of instruments and their prices, volumes, volatilities, and other values at discrete times – the core data model for Metropolis is the time series. (In Gotham, the core data model was the object graph). The time series is one type of "zero-order" objects in the Metropolis object model, one of Palantir's building blocks of analysis that form the material for more complex, higher-order objects. The instrument is another type of zero-order object, and with its instances representing named "real world objects" like companies, commodities, or currencies, and instruments are described as the nouns of the data platform. A third type of zero-order object, the metric, receives an object or objects as input and returns an object or objects as output, and metrics are described as the "verbs" of Metropolis. For example, in the Bridgewater presentation, their metric returned a new ActivityGauge time series as the aggregate of three other time series. These zero-order objects can be linked to one another and used in the composition of higher-order objects like instrument groups, indexes, date sets, or regressions.

The zero-order objects form the logic of the platform as a plane of consistency, a logic focused not just on times series as "given" data objects for examination, but as material for transformation (by metrics) into new data objects understood as aspects or measures of the initial object. (Palantir filed a patent devoted entirely to an object-oriented time series generator and its logic.) These resulting transformations can be juxtaposed and compared within the "common place" of the Metropolis platform in order to locate and describe the time-dependent value relationships between instruments as well as higher-level objects like indexes. A key feature of analysis performed in Metropolis is how these capabilities allow for the articulation of different hypotheses by allowing derived data objects to appear in the same plane as "given" data and each other—in the language of computer science, all Metropolis objects are "first class" objects.[17] According to Latour, the optical plane of consistency, created by "contrivances" like perspective and projection, is powerful not because of its realism but because it provides a site for hybrids, for "nature seen as fiction, and fiction seen as nature."[18] The Metropolis object model provides an analogous framework for the translation and transformation of time series data into new, derived data objects while preserving their relationships with the original domain.

This understanding of Metropolis, abstracted from histories of visual culture and craft, may initially appear at odds with Palantir's explicit design metaphors invoking natural language. Palantir's literature describes a given deployment's objects as elements of that organization's "analytical vocabulary," – and the object model itself, in structuring the treatment of those objects, can be said to constitute a grammar of analysis for time series.[19] In the same register, the user-created analysis components and products – custom metrics like Bridgewater's ActivityGauge, for instance – are stored as document objects in a firm's database. However, the activity of this "grammar" goes beyond common understandings of language as descriptive or

[17] Ibid.
[18] Latour, "Visualization and Cognition," 9.
[19] "Palantir Metropolis Applications."

expressive. This grammar, written in code, should be understood as “performative and processual,” producing changes in a total machine process that the human user glimpses, at higher levels of abstraction, and construes as representational.[20] In Metropolis, code appears as a variety of experimental performances that do not, properly speaking, constitute descriptions of financial instruments in terms of their properties and relationships. Rather, these performances operate on digital objects representing stocks, bonds, and currencies – in particular their time series – in order to elicit such descriptive information – often producing new time series in the process or as the ultimate result. The platform can be said to supply users with a grammar of analysis, but it also can be said to provide them with a kind of laboratory or workshop. Financial analysis appears as an experimental activity.

A Palantir blog post entitled “Substitutes for Oil” illustrates the role of time series generation as an experimental framework, showing how an analyst using the platform might identify stocks whose price movements closely track the price of oil.[21] The key to this workflow is the use of a metric *removeMarketSector* to decouple a stock’s price movements from those of its index (taken as a proxy for the market as a whole) and its industry sector. Taking an Instrument object as input, this *removeMarketSector* performs two regression calculations for that Instrument’s time series, first against the S&P 500 and then against the related sector ETF (exchange traded fund). The metric returns the regression residuals as a new time series object. Using the Instrument Explorer application, the analyst is able to integrate this metric into a filter applied across all the component stocks of the S&P 500. This filter, as written, takes the series resulting from the *removeMarketSector* (expressed as percent change day-over-day) and calculates the correlation between those series and that of the price of oil.

[20] Berry, “A Contribution Toward a Grammar of Code” (see Introduction n. 56).
[21] Nima, “Substitutes for Oil,” *The Palantir Blog*, September 6, 2009, https://www.palantir.com/2009/09/substitutes-for-oil/. (Page removed.)

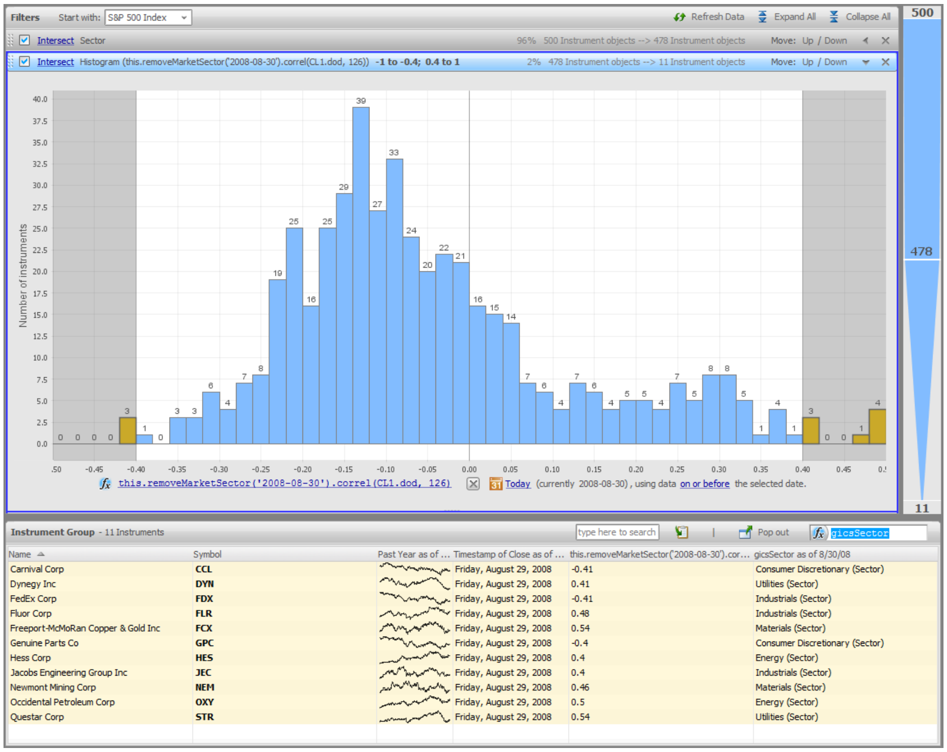

Figure 3.5: Finding correlated stocks using Instrument Explorer. From Palantir Blog post "Substitutes for Oil."

Ultimately, this work in the Instrument Explorer returns eleven stocks whose market- and sector-independent movements are closely correlated with the price of oil. To provide further evidence of relationship, the analyst creates a custom tracking index composed of these eleven stocks (as its own Metropolis object) and compares the monthly returns of that index to USO, an oil ETF.

Metropolis takes the object-oriented epistemics displayed by the Palantir Gotham platform in a new direction. With Gotham, the adding, paring, and repositioning of objects in the Graph application preserved the essential integrity of these data items. Metropolis, by contrast, introduces a focus on operations that extract and transform the properties of its data objects, generating new (but related) time series or other quantitative data to isolate or combine hypothesized properties from the original object (or, rather, its time series). The "Substitutes for Oil" demonstration, first of all, highlights a set of tools within Metropolis for subtraction or removal that produce new time series for stocks that are, hypothetically, isolated from those of the larger market. Following this isolation, the correlation metric brings the objects into proximity for comparison. Finally, a new, higher-order object is created from the combination of the eleven stocks for comparison to the oil ETF as additional evidence for the relationship or connection. The object-oriented approach extends beyond the individual analyst to the organization as a whole, as the products of analysis – for instance, the new oil-related index – can be stored as their own objects in the database and called up by other users within the organization, contributing to the entire firm's perspective on markets and its evolution. This ability to store and reuse user-generated objects and methodologies is how Metropolis realizes the Palantir's promise that the platform can "grow along with your organization's view of the world."[22]

[22] Palantir, "Dynamic Ontology" (see chap. 2 n.

With this framework in mind, we can understand the different named applications within Metropolis—Chart, Custom Metric, Date Set, Instrument Explorer, Index, Regression, Strategy, and Spreadsheet—as sub-environments staging various templates for selecting, generating, comparing, and/or visualizing time series objects; these templates can be applied in various sequences and their results can be combined and used within other parts of the analysis. Each application responds to specific knowledge problem surrounding the larger database of objects and enables the analyst to ask a certain range of questions about the data. Instrument Explorer asks which objects fit user-provided criteria. Regression asks after the degree to which the values of one series are related to the values of one or more other series. Index permits questions after the characteristics of aggregate series. Strategy performs calculations on time series objects to evaluate the earnings of instruments bought and sold according to defined criteria.[23] In the course of a workflow, these applications support not only the movement between alternative views and perspectives on markets, as in the first part of the Bridgewater demonstration, but they also create the possibility of constructing new, interleaved perspectives through the combination of objects and operations.

Resembling the structure of molecular biology laboratories examined by Knorr Cetina, Palantir's platforms present the analyst with a multitude of "little instruments" for the presentation, manipulation, and analysis of digital traces.[24] The time series sits squarely at the center of Palantir's object-oriented, experimentalist software environment, with the hope being that a given assembly of analytical "building blocks" produces an ultimate set of signs – visual, numerical – that can "stand for the invisible, phenomenal realities" of capital markets.[25] With respect to the Metropolis object model, the search for an underlying financial reality and the specific qualities of instruments first relies on a thickening of the field objects and signs. This is not an analytical regime of detached observation but one of interactive processing where, as Knorr Cetina says of the molecular biology laboratory, "objects are decomposable entities from which effects can be extracted through appropriate treatment; they are ingredients for processing programs […]."[26] The financial analyst takes on the mantle of the experimentalist, varying procedures to transform and breakdown these quantitative data objects, deploying an embodied self (both the "sensory" and the acting" body, as Knorr Cetina terms them) to proliferate numbers and charts that eventually "cut through the appearance of signs until the observer arrives at phenomenal reality, not visible from the surface of the image."[27]

These explorations in Metropolis reflect the deep theme introduced by the Bridgewater speaker, that of undiscovered linkages within and across global asset markets. Moreover, Bridgewater presents their choice of Palantir as a result of the platform's compatibility with this view of markets and investment, a view that sees "the world as a machine… the economy as a machine." The avowed goal of financial analysis, then, is to understand this machine in detail and grasp its "deep whys." In this view, asset prices are signals of an underlying economic process. Any assertion of the existence and quality of a relationship between instruments and/or higher-order objects, like indexes and sectors, is made with reference to relationships between their associated time series. Sometimes these data objects are "given" but, more often, they are generated by the analytical process – in Metropolis, a process driven by the human user. In the

[23] "Palantir Metropolis Applications."
[24] Knorr Cetina, *Epistemic Cultures*, 85 (see Introduction n. 33).
[25] Ibid., 104.
[26] Ibid., 37.
[27] Ibid., 101.

work within and between the Metropolis applications, certain sequences are geared toward isolation or de-coupling of time series objects – for instance, the selection of groups of instruments from a broader index, or regression analysis to decouple a stock from wider market movement. With a narrower focus on a limited set of instruments and/or metrics, the analyst is prepared to spot, if present, the systematic relationships among financial objects.

In the words of the Bridgewater researcher: "when we do our research, what we're looking for is the linkage, the ability to explain the machine, to be able to say how that gear connects to that gear, here's why that produces a certain output." From Bridgewater's perspective, the value-added of the Metropolis platform is its ability to support analysts in spotting connections, and the "Substitutes for Oil" post exemplifies this work of producing and systematizing hypotheses about relationships among instruments, work that centers on the manipulation and transformation of objects. Analysis in Palantir Metropolis involves variable sequences of selection, separation, imaging, calculation, combination, and revision with the hope that, in the end, visual and numerical indicators from these investigations serve to illuminate the invisible momenta of capital markets. Each Metropolis application tends to be organized around a specific set of questions with regard to the selection, manipulation, or representation of relevant time series objects, and the Metropolis platform as a whole allows the answers to these questions to be sutured together into a more comprehensive understanding of market phenomena – an environment for reverse engineering the market machine.

4. Strategy and Search: *Speculative Pasts and Uncertain Futures*

Ultimately, actors in financial markets are not content to simply discover a past relationship among market objects – the goal is to profit from knowledge of the connection, that is, to exploit it as an opportunity in the future. The problem, then, is to transform an insight about such a past relationship into a specific formula, a strategy, for making future decisions about transactions: under what conditions should the investor buy or sell which assets? In other words, the investor needs to have methods of "folding the future back into the present" (a present which includes knowledge of prices up until this moment).[28] According to Beunza and Garud, the focus of orthodox economics on calculative decision-making in situations of complete, probabilistic risk knowledge has resulted in inadequate attention to how market actors cope with Knightian uncertainty and "the imperative to decide with a significant but limited knowledge of the world."[29] In trading contexts, one such method is called backtesting, the specification and evaluation of a trading strategy using historical data. In essence, backtesting constructs a hypothetical past in an effort to characterize this future, presuming a measure of resemblance in order to project the probabilistic risk knowledge, gleaned from that past, onto an uncertain future.

This work of converting uncertainty into risk, left unaddressed in the previous demonstrations, is the focus of Metropolis's Strategy. An environment for the development of detailed trading strategies in algorithmic form and the backtesting of those algorithms with varying parameters, the Strategy application is a site for the specification, extension, and potential validation of the general market insights seen in the previous examples. The Strategy

---

[28] Daniel Beunza and Raghu Garud, "Calculators, lemmings, or frame-makers? The intermediary role of securities analysts" in *Market Devices*, eds. Fabian Muniesa, Yuval Millo, and Michel Callon (Oxford: Blackwell, 2007): 13.
[29] Ibid.

application extends the interactive processing characteristic of the Metropolis platform to the pursuit of opportunity and the construction of courses of action with understood probabilities of loss or gain.

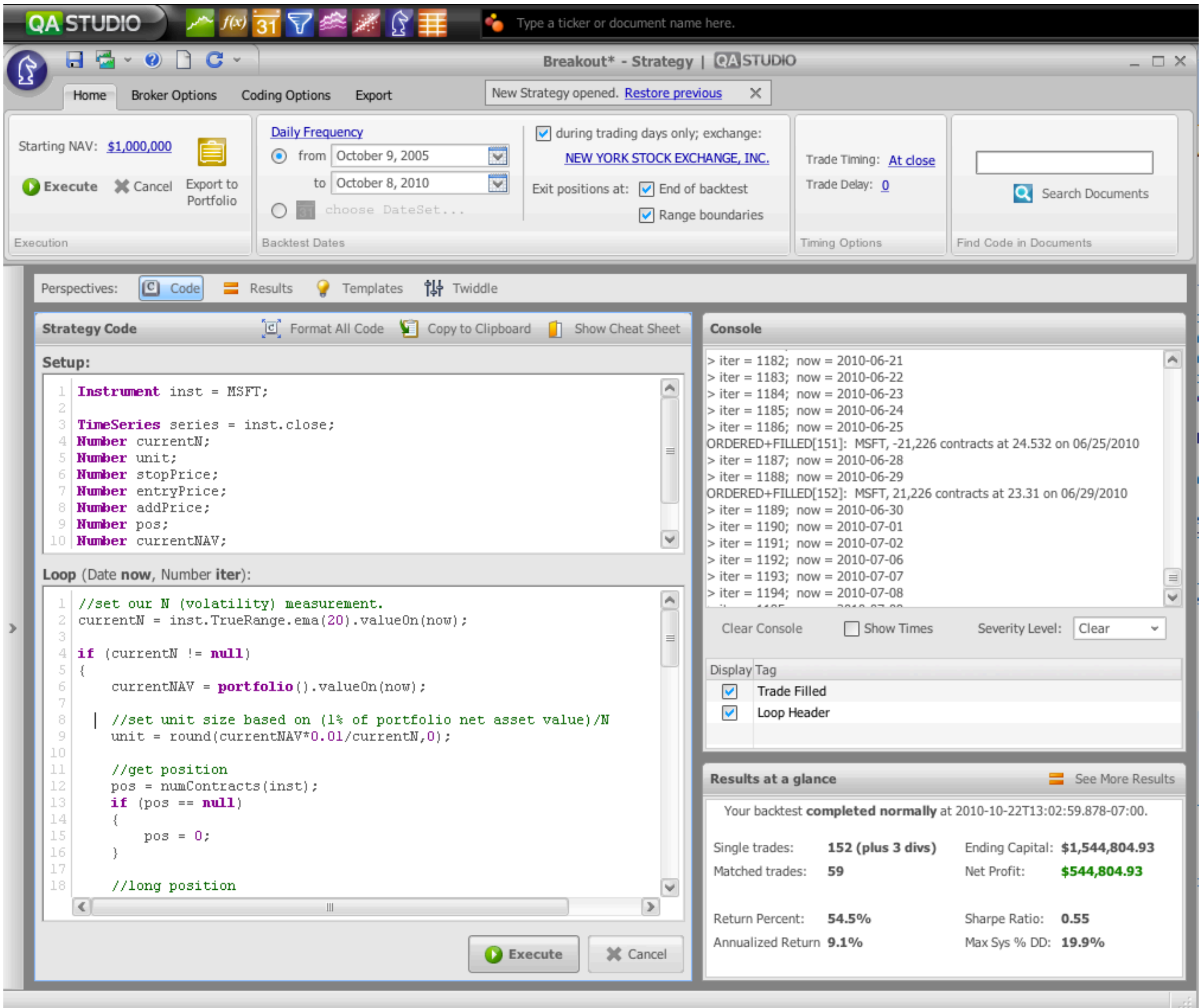

Figure 3.6: Screenshot of Strategy application, from Palantir Blog post "Turtle Breakout Trading Strategy Simulation."

In the upper part of the Strategy application interface (Figure 3.1) are global parameters, where the user specifies an initial NAV (net asset value), a range of dates and information regarding the frequency of execution, timing. Within the Strategy "Home" tab, there are four additional tabs or perspectives: Code, Results, Templates, and Twiddle. The first among these, the Code perspective, contains text entry boxes for setup and loop code, with the former defining variables for use in the latter. The loop code, as one might imagine, is executed according to the frequency and date ranges specified. A user inputs the strategy code directly using HHLang, invoking built-in or custom metrics and other modes of evaluation to be met before an action is taken. A console pane logs the results of the execution, displaying the current date, iteration, and any trades executed. Finally, a "Results at a glance" pane displays aggregate measures like the number of trades, starting and ending capital, net profit, Sharpe Ratio (a measure of risk-adjusted return), and return percentages (both total and annualized). A larger set of statistics, charts, and information on individual trades resulting from the strategy code are displayed in the more detailed Results perspective.

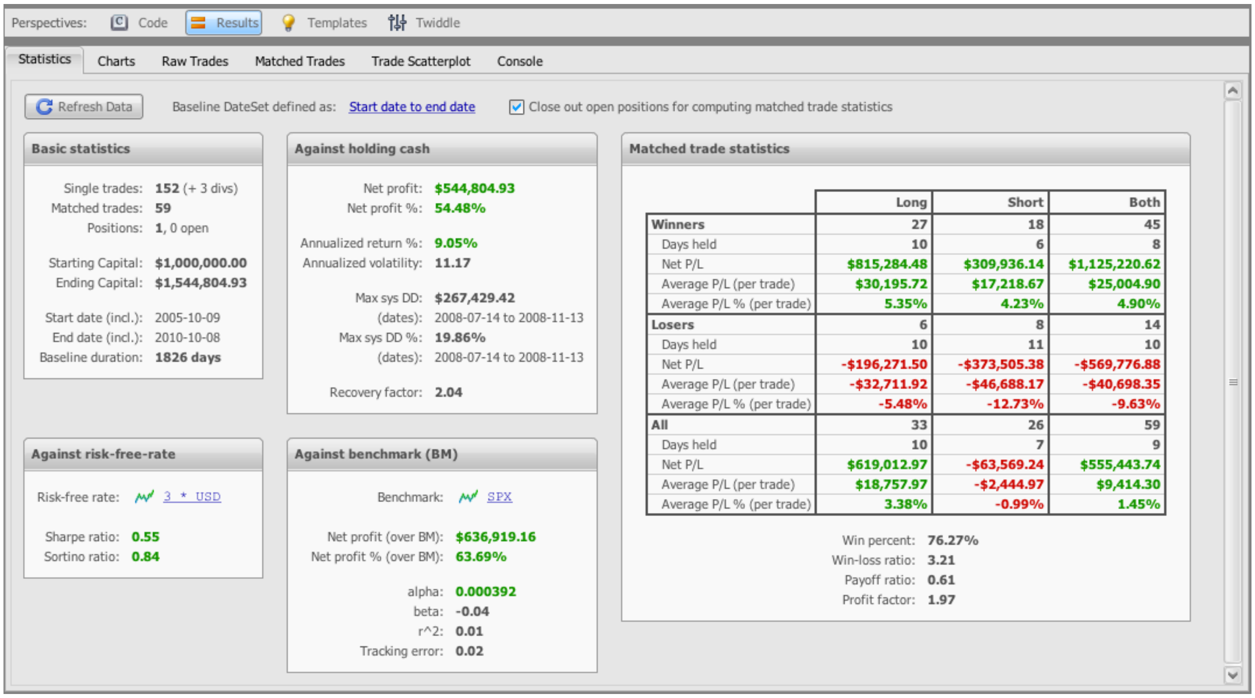

Figure 3.7: Detailed results from a recent Strategy test. From Palantir Blog post "Turtle Breakout Trading Strategy Simulation."

A number of posts on Palantir's now-defunct analysis blog explore various market hypotheses and/or trading strategies using the Metropolis platform, often displaying the ways in which Metropolis allows for the creation of metrics to isolate desired aspects of market measures, visual and statistical evaluations of their correlations, and procedures for coordinating those aspects to produce systematic trading approaches. I will examine one post here that walks through an evaluation (and improvement) of a trading strategy known as "the Fed model,"[30] which is based on the hypothesis that the yield of equity markets, when offset by a risk premium, should be equivalent to that of a 10-year government bond – in other words, the difference between these two measures should remain constant.[31] If there is indeed a fundamental economic link between these two measures, any large divergence between these measures would mark an opportunity for profitable trades in anticipation of their return to a constant difference. For instance, if earnings in equity markets increase without a corresponding increase in share prices while bond yields remain constant, the Fed model implies that stocks are undervalued, and one would expect their prices to eventually rise as investors move back toward these assets with their more favorable rates of risk-adjusted return.

[30] The Fed Model hypothesis appeared in the Federal Reserve Board's Humphrey-Hawkins report of July 1997, as well as in the journal article by Fed research staffers Joel Lander, Athanasios Orphanides, and Martha Douvogiannis, "Earnings Forecasts and the Predictability of Stock Returns Evidence from Trading the S&P," *The Journal of Portfolio Management* 23, no. 4 (Summer 1997): 24-35.
[31] Nima, "The Fed Model," *The Palantir Blog*, July 15, 2009, https://palantir.com/2009/07/the-fed-model. (Page removed.)

The resulting code in the Strategy application compares the actual value of the S&P 500 index and an implied S&P value derived from "a ratio of the S&P 500 earnings yield and the 10-year [United States] Treasury [bond]." This strategy "enters a trade when the difference of the two [time] series reaches two standard deviations above its 50 day moving average, and exits the trade when it moves back to its moving average," along with integrating a stop-loss at 10% of net asset value. A simulation (or backtest) of this initial strategy over a historical eight-year period shows what the Fed Model post's author sees as relatively poor performance. The Twiddle perspective within the Strategy application can be used to compare the effects of varying certain parameters – in this case, the number of days in the moving average and the number of standard deviations for the trading threshold.

Figure 3.8: Twiddle tool within the Strategy application. From Palantir Blog post "The Fed Model."

These results are shown in a color-coded matrix that highlights combinations with the highest Sharpe Ratio (a measure of risk-adjusted rate of return), but, again, the author sees no satisfactory combination. The Twiddle perspective, in its presentation of a matrix showing parameter combinations and their calculated results, ultimately moves decision-making into a common visual space where a field of related options can be juxtaposed and compared.

Unsatisfied with the results thus far, the analyst proposes a revision this strategy based on their hypothesis that "one of the main problems with the Fed Model is that it assumes that the credit quality of the S&P 500 matches the credit quality of the 10-year Treasury." As a result, it seems that the yield of 10-year Treasury bonds does not, indeed, generally track the forward earnings yield of the S&P 500. With this in mind, the user opens the Instrument Explorer, an application for the specification of advanced searches for instruments within the Palantir deployment. The investigative turn here is not to abandon the strategy but to revise it using a potentially more relevant benchmark – that is, to identify a new candidate for one of the objects of measure and correlation at the center of the strategy. If "[u]sing that bond index, we can better predict the value of the S&P 500," the potential for divergences signaling a fundamental over- or undervaluation of stocks may be increased.

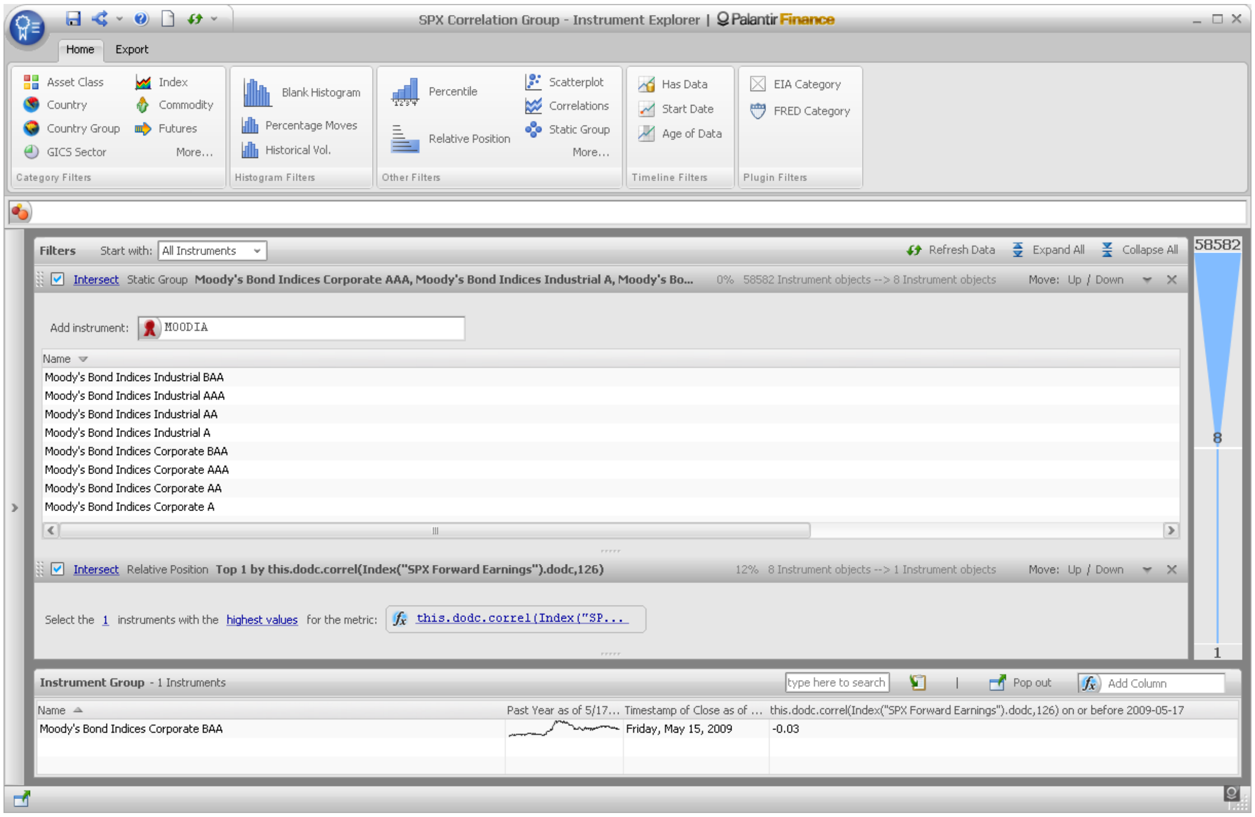

Figure 3.9: Instrument Explorer filtering for the bond index with the highest correlation to the S&P 500 forward earnings yield index. From Palantir Blog post "The Fed Model."

The analyst turns to Instrument Explorer application to search for a bond index more closely correlated with S&P 500 forward earnings yield. The analyst selects all Moody's Bond Indices for corporate bonds and sets filters for the instrument with the highest value for the metric entered, in this case its correlation with the S&P 500 forward earnings yield, which returns, for the past year (as of May 17, 2009), the Moody's index for corporate BAA rated bonds.

Using the Index application, the analyst then creates a custom index that, on a quarterly basis, chooses the bond index best correlated to the S&P 500 yield. The analyst can view the resulting quarterly choices and their corresponding yields on a chart within the Index application. The analyst takes this another step, creating an index that tracks the Z-Score of the difference between quarter's bond index yield and the S&P 500 forward earnings yield. Finally, the analyst creates a revised Strategy based on this model, using the Z-score metric as a threshold for entering and exiting trades. In the end, a backtest over the same period as the first Strategy shows a much higher annualized return (34.3% versus 11.7%). Based on these results, the Fed Model author concludes that "the Fed model can be greatly improved upon by introducing a more sophisticated model for the equity market credit quality." Moreover, the user was able to explore, construct, and integrate this new model using only the objects and interactions available with the Metropolis platform. As with the other workflows detailed in this chapter, the elaboration of the trading strategy relies on the platform's capacity to render hypotheses, fictions and hybrids, together in the same environment or plane of consistency.

The Fed Model demonstration attempts to show how Metropolis can enable open-ended exploration of a trading strategy, an exploration resulting in a more nuanced and higher performing revision of well-known trading approach. Central to the post is the Metropolis platform's ability to display results, to inspire searches for alternatives, and the ability rapidly search for components and construct a more complex object within the environment – in this case, a single bond measure (the yield of 10-year US Treasuries) is replaced by a dynamically rotating index object that shifts quarterly based on a correlation metric. Finally, an evaluation of the revised strategy seems to confirm the analyst's hunch, the lightly articulated hypothesis, that the original rendering of the strategy provoked. While the Bridgewater demonstration emphasized the macro-level optics of the Metropolis platform, the Fed Model demonstration presents a more in-depth example of the laboratory character of the platform in its orientation toward structured variations on procedure—in other words, experimentation.

Fiction, hybridity, and craft are the dominant themes in these demonstrations, and the production of knowledge is articulated with the activity of crafting fictions and hybrids using the tools given and within the consistent plane of the Metropolis platform. The knowledge and decisions that result imply an alternative ethics of objectivity, one not derived from detached observation or logical deduction but from the engagement of objects, instruments, and the user. The Strategy application is a capstone moment for the processes of construction and validation, a creative process that in creating the strategy also creates its visibility, even though it may be retroactively characterized as a course of action that existed *a priori* and only needed to be, as it were, "found."

Design and Search

The analytical rhythms characteristic of laboratories and workshops that run through these Metropolis workflows trouble understandings of knowledge and problem-solving, even in the highly quantitative domain of finance, as primarily calculative and deductive. The appearance of craft, construction, and design in this context are not incidental features of the Metropolis platform to be pitched to firms who happen to prefer human-driven financial analysis. Herbert Simon claims that activities closely associated with the practice of design are, more so than calculation, the primary constituents of problem-solving processes *in general*. The reason for this is the high "cost" of search with respect to determining the optimal course of action in response to a problem. Ideally, one would exhaustively derive, compare, and choose from the set of "possible worlds" as representing valid transitions from the current situation to a desired one. But even a well-structured and limited problem like a chess game includes an exponential number of possible legal board configurations intervening between the beginning and the end of the game – Claude Shannon estimated the lower bound of chess's game tree complexity (the number of potential end points for a given first move) at 10^{120}. Given current technologies, the search costs for chess (and many other problems) are too high in terms of time, energy, or both – all alternatives, all possible worlds, cannot be practicably considered.

For Simon, then, design is a kind of "bottom up" approach that controls the cost of search in problem-solving processes. First, design shifts the goal of problem solving from pursuit of the optimal choice to pursuit of a satisfactory choice. Second, designers go through cycles of generation, elaboration, and testing of alternatives ("possible worlds"), often using heuristics and experience to guide them in this process of limited search.[32] These cycles involve moves between problem representations coupled with mechanisms for noticing interesting features (or sub-problems) and recalling potentially relevant responses from long-term memory.[33] For Simon, the characteristic procedures a designer/problem-solver uses to generate and test alternatives,

[32] Herbert Simon, *The Sciences of the Artificial* (MIT Press, 1996), 117.
[33] Simon, "Ill Structured Problems" (see chap. 1 n. 42).

which can also be understood as procedures for limiting and exploring the search space, are the determinants of style and the differences among satisfactory solutions produced by different designers.[34]

Financial analysis, then, as a form of problem-solving is characterized by what Beunza and Stark call "the challenge of search and experimentation" with respect to potentially profitable future transactions, with respect to opportunities.[35] In response to this challenge, Metropolis partakes of a style called heterarchical search. Girard and Stark, in their analysis of web design firms of the early dot.com era, observed modes of organization "characterized by minimal hierarchy and by organizational heterogeneity," as well as "relations of interdependence" among different units within the firms.[36] Heterarchical organization, according to Girard and Stark, responds to these firms' need to "maximize their capacity to recognize opportunities and realize their promise, not only by exploiting their immediate benefits but also by exploring them as openings to new opportunities." For firms in business landscapes characterized by "dizzying impermanence," heterarchy is a successful approach because it creates "an organizational space open to the perpetual redefinition of what might constitute an option." The role of heterogeneity is key, as what makes heterarchy powerful principle of search and organization is its capacity to "keep multiple regimes of worth in play and to exploit the resulting ambiguity."[37]

Beunza and Stark find these principles at play in finance in the context of a Wall Street arbitrage trading room where they observe "a kind of laboratory in which traders are engaged in a process of search and experimentation."[38] The crucial moments in the recognition of opportunities involve the close proximity of the different "desks," clusters of traders and devices engaged with different kinds of markets and trades. This proximity contributes to new interpretations of the properties of instruments due to the constant presence of alternative categorizations.[39] "It is the friction at the interacting overlap that generates productive recombinations" behind the innovative and successful arbitrage trade and frame the ultimate recognition of a new set of *connections* along with a course of action—knowledge of an opportunity.

I would maintain that in Metropolis, as in the case with Gotham, the uniqueness of analysis in Palantir's product is not primarily due to its low-friction extensions of the calculating human mind. In any given moment, characteristics of Licklider's man-computer symbiosis appear in these demonstrations of the Metropolis platform – rapid search for data objects, autocompletion of typed commands, fast calculation and visualization, and other "clerical" obstacles to deep thought. But the moments of "insight" and recognition are more akin to the "crazy ballet" a Palantir speaker ascribes to the performance of "ZacksS," the assemblage of two amateur chess players, three consumer computers, and four chess engines that defeated both purpose-built chess computers *and* computer-aided grandmasters in the 2005 PAL/CSS freestyle chess tournament.[40] Cases like these suggest that certain kinds of friction and difference are

---

[34] Simon, *Sciences of the Artificial*, 130.
[35] Daniel Beunza and David Stark, "How to Recognize Opportunities: Heterarchical Search in a Trading Room" in *The Sociology of Financial Markets*, eds. Karin Knorr Cetina and Alex Preda (Oxford University Press, 2005): 85.
[36] Monique Girard and David Stark, "Distributing intelligence and organizing diversity in new-media projects," *Environment and Planning A* 34 (2002): 1934-5.
[37] Ibid., 1934, 1936.
[38] Beunza and Stark, "How to Recognize Opportunities," 85.
[39] Ibid., 95.
[40] Palantir Technologies, "Human-Computer Symbiosis" (see chap. 1 n. 6).

productive in finding the right move or the right trade. With the time series as the concept underlying each application (as with the object graph in Gotham applications), Metropolis emphasizes movement between perspectives on a market phenomenon, presenting the analyst with a number of "what if" choices for the selecting, visualizing, generating, and evaluating potentially relevant time series objects. In these iterative cycles, some moments call for the proliferation of objects while others call for paring. Other moments call for sustained attention – visually and algorithmically – to a very few in efforts to transform general hunches into specific insights. As epistemic mediators, the objects situated within Metropolis are not oriented toward a single diagrammatic revelation of hidden properties.[41] Rather, the objects and the Metropolis environment present an opportunity for choice and interruption within these cycles of analysis and a chance shift between different representations. These shifts characterize a software possibility space resembling a design environment, a space that subordinates theory and calculation within the plane of consistency to court productive differences and heterarchical search.

5. Performance and Incorporation

"Bottom up," design-based approaches to problem-solving could be seen as stopgap measures, workarounds for the practical limits of human cognition and technology that make do with finding satisfactory solutions in an economical near future instead of optimal ones in an impossibly distant or expensive one. In the context of financial markets, such practices would appear as more or less effective but primitive means for selecting strategies from the inaccessible but ultimately "real" space of all possible strategies. This understanding, however, is increasingly troubled by recent scholarship at the intersection of science and technology studies and economic sociology. Studies of the construction of markets, their continued production and reproduction by the actors, devices, and other infrastructures "in" and/or "around" them, change the stakes of interpretive market devices like Palantir's Metropolis platform. In particular, studies of the performativity of economic models introduce questions as to what degree connections "observed" in and between financial markets actually follow the dissemination of such models, suggesting that the reality "described" is in fact produced. In other words, we must consider the possibility that the effects of these devices, the strategies they help "design," and the styles of search they display in the course of their activity, are not purely epistemological in character, taking place only "in the heads" of market actors. Interpretive market devices also participate in the ontological processes that constitute the market as a set of evolving and contingent performances.

Attention to these kinds work on and within markets has been shaped by recent interactions between science studies and economic sociology, and the import of such inquiries is often framed by Michel Callon's provocative claim regarding the performativity of economics. Though sharing the initial assessment that markets are not spontaneous manifestations of natural economic energies, Callon then departs from orthodox economic sociology and proclaiming that rather than seeing markets and economies as embedded within society, scholars should study them as embedded within economics. "Economics," as Callon uses it for these purposes, is meant in a "broad sense" and encompasses various market-related tools of measurement,

[41] Magnani, *Abductive Cognition* (see chap. 1 n. 75).

standardization, exchange, and, more controversially, economic theory.[42] For Callon, the rational actor of economics, the calculative *homo economicus*, is a temporary character that emerges from an ensemble of tools he calls calculative agencies; these agencies allow for the (always incomplete) framing and disentangling of persons and goods from other contexts (social, cultural) and their appearance in the impersonal spaces of exchange. "To construct a market transaction," he writes, "to transform something into a commodity, and two agents into a seller and a consumer, it is necessary to cut the ties between the thing and the other objects or human beings one by one."[43] Initially posited as a supplement to the insight (and consequent paradox) of Mark Grannovetter's social network analysis of economic activity, Callon describes framing as the necessary process of drawing "a clear and precise boundary… between the relations which the agents will take into account and which will serve in their calculations and those which will be thrown out of the calculation as such."[44] Framing is fundamental to economic behavior in that it allows actors to determine which parts of the network "count" for a transaction and which do not.

In terms of the resources for the framing, disentanglement, and the construction of markets, Callon acknowledges the "material, procedural, legal and monetary elements which facilitate the framing of the construction of the space of calculability," but he ultimately wants to turn our attention to "a capital, yet rarely mentioned, element: economic theory itself."[45] Callon claims "this performation of calculative agencies—i.e., of the economy by economics—is largely carried out through the intervention of professional economists."[46] Economists do not just make the models that describe or inform the activity of market actors. Economics, as a repository of tools that equip social actors and transform them into the calculative agencies of market contexts, is said to *perform* economic phenomena themselves, in the sense described in J.L. Austin's work on speech acts.[47] Callon's performativity thesis is an ontological thesis, one that suggests that economic modeling is not (or not only) a social activity of description and justification for certain kinds of behavior that we call "economic," but a form of effective action caught up in the *actual* making of markets and economic agents.

Callon's performativity thesis is, in part, a consequence of the principle of symmetry at the heart of actor-network theory (ANT). Previous applications of this principle have focused on the inclusion of nonhuman actants like scallops and peptides along with human scientists in sociological stories about the making of facts, science, and technology.[48] In Callon's reading of economic sociology, however, it is the role of human economists (as scientists) that is overlooked in stories of how markets are made. Callon's ANT commitments, according to Peter Holm, result in an initial insistence that "markets really exist and make a difference in the world" and cannot be reduced to social relations.[49] Following this, Callon argues that economists'

---

[42] Michel Callon, "Introduction: the embeddedness of economic markets in economics," in *The Laws of the Markets*, ed. Michel Callon (Oxford: Blackwell, 1998), 2.
[43] Ibid., 19.
[44] Ibid., 16.
[45] Ibid., 22.
[46] Ibid., 30.
[47] J.L. Austin, *How to Do Things With Words* (Cambridge, MA: Harvard University Press, 1962).
[48] See: Michel Callon, "Some elements of a sociology of translation: domestication of the scallops and the fishermen of St Brieuc Bay," *The Sociological Review* 32, no. 1 (1984): 196-233; and Bruno Latour and Steve Woolgar, *Laboratory Life: The Construction of Scientific Facts* (Beverly Hills: Sage, 1979).
[49] Peter Holm, "Which Way Is Up on Callon?" in *Do Economists Make Markets?: On the Performativity of Economics*, eds. Donald MacKenzie, Fabian Muniesa, and Lucia Siu (Princeton: Princeton University Press, 2007): 227.

descriptions and prescriptions (among other things) should not be read purely as *post hoc*, ideological justifications of certain social configurations called "markets," but as effective elements in the actor-networks that make these markets real. In Callon's view, markets are what sometimes result from the co-location and coordination of calculative agencies, agencies that are formatted to a great degree by the work of economics. If, at one point, ANT was primarily focused on the co-production of Nature and Society in laboratories and other technoscientific contexts that resulted from alliances and oppositions between human and non-human actants, Callon's thesis extends this approach to the co-production of Economy and Society in the making of markets that results from interactions between laws, devices, actors, models, and—importantly—economists.

Callon's diagnosis of presumptions of the social embeddedness of markets generates a number of productive avenues of research and theorization regarding the configuration of economic actors in various market contexts, the different subspecies of *homo economicus*. His insistence, however, on the role of professional economists can lead to analytical accounts that lack the contingency and complexity characteristic of many contemporary studies of science and technology, and especially those inspired by ANT. One cause of this difficulty, according to Judith Butler's critique of Callon, results from the origins of the concept of performativity in the J.L. Austin's work and the pride of place given to illocutionary acts performed by "sovereign" speakers therein, Austin's many instructive (but not exhaustive) cases where the saying makes it so.[50] While ontological effects can and do proceed from economic theory, Butler claims that those effects would often be better characterized as perlocutionary effects, dependent as they are on "good circumstances" such as the manner of uptake and iteration by other actors.[51]

I see a response to Butler's call for a consideration of the perlocutionary model in Donald MacKenzie's account of *incorporation* in his story of derivatives markets in *An Engine, Not A Camera*. For MacKenzie, what he specifies as the "effective performativity" of an economic model is contingent on its integration into the beliefs of market actors and, perhaps more importantly, its incorporation into "algorithms, procedures, routines, and material devices," a form of incorporation that "can have effects even if those who use those who use them are skeptical of the model's virtues, unaware of its details, or even ignorant of its very existence."[52] These different forms of incorporation—linguistic, technical, and legitimatory—guide MacKenzie's examination of markets and their infrastructures as contingent achievements of technics, rhetoric, morality, and politics.[53] This vocabulary of incorporation is, I believe, more specific and ultimately responsive in the context of a critical analysis of markets. It allows inquiry to proceed concretely from explicit models to what forms of incorporation may result, and it can also ground an examination of new and existing devices and infrastructures to discover, through their design and deployment, what models, both explicit and implicit, are already incorporated.

Caitlin Zaloom, for instance, details the asceticism pronounced by futures traders in Chicago, a personal spiritual discipline with respect to one's memory, emotions, and the non-market occurrences of everyday life.[54] To deal with the deluge of quantitative data, the traders

---

[50] Judith Butler, "Performative Agency," *Journal of Cultural Economy* 3, no. 2 (2010): 153.
[51] Ibid., 151.
[52] Donald MacKenzie, *An Engine, Not a Camera* (Cambridge, MA: MIT Press, 2006), 19.
[53] Ibid., 250.
[54] Caitlin Zaloom, *Out of the Pits: Traders and Technology from Chicago to London* (University of Chicago Press, 2006), 127-8.

Zaloom observes engage in constant chatter, offering snippets of narrative and hypotheses to situate and explain the price movements.[55] These snippets of discourse we might see as market devices, tools that make traders by keeping them attuned to and oriented toward their common object, one aspect of the microstructures Karin Knorr Cetina and Urs Bruegger argue constitute global financial markets.[56] Other tools make their "device" character much more explicit, tickers, telephones, and, of course, screens. Knorr Cetina and Barbara Grimpe, observing foreign exchange trading floors at investment banks, note the "traders' eyes are glued to these screens even when they talk or shout to each other."[57] These screens are the visual interface for scoping systems where "the market constitutes itself in these produced-and-analyzed displays," making traders in part by presenting them with current information and their proper space of engagement, of knowledge and action.[58] Palantir Metropolis belongs among this category of market devices, in particular those interpretive devices that attune and sensitize traders and analysts to the various data and indicators of financial markets, supporting their ongoing endeavors to recognize and exploit new opportunities suggested therein. A survey of these devices and infrastructures reminds us that economic theory proper is not the only source of "models" in markets, with computer interfaces, market chatter, and other items supplying important but perhaps less determinate fragments of strategy, worldview, and calculative capacity.

Along with accounts that proceed from the discourse of the economists, incorporation provides a framework for accounts that begin with market devices as cases and lay out the connections, dependencies, and the resulting possibilities–possibilities of success, failure, growth, and crisis–that constitute the heterogeneous spaces of market activity. From this perspective, Palantir Metropolis as an analysis platform, incorporates and presents to users a set of different and open-ended templates for the calculation, visualization, and interpretation of market data, and it does so within a logic that connects each of them not just theoretically but interactively. Rather than embodying a strict prescription of the relevant elements and their quantitative relationships within a market, Metropolis displays how actors' interpretations may be developed, revised, and stitched together using a variety of tools. Traders and analysts are not limited to "performing" economists' pronouncements; rather, they have the option of using those incorporated models as part of a larger, heterogeneous possibility space for understanding market activity, recognizing opportunities, and completing transactions.

Economists may be said to make markets, but any performation depends on the participation of larger ensemble of designers, programmers, and others who materialize their models for use as devices in a wider context of perception, interpretation, and action. Moreover, the "truth effect" traditionally associated with performativity is only one possible kind of effect that can result from the incorporation and dissemination of interpretive market devices. A central case in Donald MacKenzie's *An Engine, Not a Camera* suggests a number of different modalities for such effects. Though MacKenzie does present the Black-Scholes-Merton option pricing model as a candidate for the kind of prescriptive performativity (what he calls Barnesian performativity) that Callon's initial thesis highlights, the bulk of MacKenzie's analysis speaks to

---

[55] Ibid., 156.
[56] Karin Knorr Cetina and Urs Bruegger, "Global Microstructures: The Virtual Societies of Financial Markets," *American Journal of Sociology* 107, no. 4 (2002): 905-950.
[57] Karin Knorr Cetina and Barbara Grimpe, "Global Financial Technologies: Scoping Systems That Raise the World" in *Living in a Material World: Economic Sociology Meets Science and Technology Studies*, eds. Trevor Pinch and Richard Swedberg (Cambridge, MA: MIT Press, 2008), 161.
[58] Ibid.

the importance of the material life of that model (in the form of sheets sold to options traders).[59] Chicago traders' investment (via a subscription service) in these sheets indicated their willingness to entertain and ultimately accept Black-Scholes-Merton as a useful device for observation and interpretation in their market context. Eventually, MacKenzie argues, prices within options markets approached were well described by the model from the mid 1970s to the summer of 1987.[60] But incorporation and dissemination appear here as critical moments of the performation of markets and the effects of economists' models.[61] There is no abstract traffic in ideas, and here economic thought would intersect with questions of design as the production of a material epistemic mediator, a tool for thinking, becomes relevant to the essence of options markets.

But while the Black-Scholes-Merton model was conceived as a theory for the pricing of options, the logical and mathematical work behind it entailed other important observations about the valuation of stocks and options and the relationships between the two. Elements of the interpretive logic in the Black-Scholes-Merton model were later incorporated into a *new* market product in the 1980s called portfolio insurance, made popular by a firm started by Berkeley scholars Hayne Leland and Mark Rubinstein, along with John O'Brien in the early 1980s.[62] Portfolio insurance combined key observations from the Black-Scholes-Merton model, prominently the possibility of mirroring the payoff of an option by continuously adjusting a position in the underlying stock. This concept of a 'replicating portfolio,' implied that, conversely, an investor could use option theory to "manufacture" a put – an option to sell the portfolio at a predetermined price – and hence create a floor price. As portfolio insurance did not constitute any sort of legal obligation or policy, the trio's preferred term was "dynamic asset allocation." Eventually, Leland O'Brien Rubinstein Associates, Inc. (LOR) implemented their portfolio insurance primarily using S&P 500 index futures (introduced on the Merc in 1982).[63]

On October 19 ("Black Monday") and October 20, 1987, global stock markets fell significantly, with the Dow Jones Industrial Average losing over 20% of its value. While the precise causes of the 1987 Black Monday crash are difficult to trace – it was the subject of an extensive investigation by a presidential task force led by Nicolas Brady – MacKenzie does cite a "plausible set of mechanisms" pointing to a potential role for portfolio insurers in the sell-off.[64] These mechanisms, in my view, underscore the range of effects and interactions that can emerge from the apparently compatible and necessarily partial interpretive logics and devices that emerged from the Black-Scholes-Merton model.

An unexpected drop in stock prices prompted portfolio insurers, based on their quantitative models, to advise their clients to sell stock index futures – this could be seen as an unreflective, mechanical action, a feature of the portfolio insurance product. This action, however, occurred at a large enough scale to appreciably push down the price of index futures with respect their theoretical price given the underlying index. Normally, index arbitrageurs, enabled by their knowledge of option theory, would step in and aggressively purchase the now undervalued index futures and short-sell stocks, correcting the discrepancy.[65] October 19, 1987, however, saw breakdowns in the material infrastructure to support this arbitrage, with software

---

[59] MacKenzie, *An Engine, Not a Camera*, 162-3
[60] Ibid., 272.
[61] Ibid., 19.
[62] Ibid., 179-81.
[63] Ibid., 183.
[64] Ibid., 191.
[65] Ibid., 186-7.

problems, network delays, and the inability of printers to keep pace with the volume of sell orders at the New York Stock Exchange.[66] As a result, the price discrepancy between index futures and the associated index continued to widen, posing a dilemma to portfolio insurers and other investors (also enabled by interpretive logic of option theory), that of which set of prices truly indicated the state of equity markets and how much and how quickly one should be selling.[67]

The interpretive logic of the Black-Scholes-Merton model, the model at the core of portfolio insurance, enabled a set of market actions, in this case, the purchase and sale of index futures to hedge the equity portfolios of certain investors. This same logic would also support the response of arbitrageurs to any major price divergences between the futures and stock markets, which, when linked with the material market infrastructure for updating prices and executing trades, resulted in a functional interconnection (but not identity) between the futures market in Chicago and the New York Stock Exchange. And when this material infrastructure failed so did the interconnection, and a widely distributed and oft-relied on interpretive logic became suddenly, critically ambiguous – were stock prices artificially high, or were futures prices artificially low? The Black-Scholes-Merton model, in various embodiments and contexts, had made sense of these two things together, and now what once made sense no longer did.

The precise role of portfolio insurance in the 1987 crash is not clear, but MacKenzie's description suggests a more expansive conceptualization of the ontological effects of interpretive market devices deployed at scale. The modes of performation at work here are not easily abstracted from concepts speech and discourse, but rather evoke a vocabulary of performation reminiscent of signal processing—circuits for coupling, decoupling, transmission, amplification, isolation, tuning, and sensitization. This inflection of the concept, ultimately, places an emphasis on how market devices, as the material incorporations of models and designs (from economists and others), act as tools for interpretation and action, affecting market ontologies by modifying the field of possibilities for market actors. Performation is not only about ways of making things "true" but about how tools of knowledge create the potential for effects like interconnection, disconnects, feedback, and amplification. Part of this potential lies in the structures and mechanisms that propagate models and make them effective in different practical contexts, the virtual societies and sociotechnical microstructures that run parallel to markets.[68] A related investigation by MacKenzie into the 1998 collapse of Long-Term Capital Management (LTCM) traces how the proliferation of incomplete imitations of LTCM's successful investment strategies created a vulnerable but unrecognized "superportfolio" across firms and capital markets, with structures of dependency only becoming evident in the catastrophic wake of what should have been a somewhat minor financial event.[69] In the end, the prostheses of market rationality straddle these apparently heterogeneous domains, creating overlapping communities of interpretation and networks of cognition, and contributing to aggregate phenomena whose causal structure crosses ontological boundaries.

Metropolis as a market device incorporates an interpretation layer, not a single interpretive logic but an on-screen possibility space for staging and comparing different interpretive and evaluative logics, a co-location of search and evaluation heuristics. The

---

[66] Ibid., 185.
[67] Ibid., 187.
[68] Knorr Cetina and Bruegger, "Global Microstructures."
[69] See chapter 8 of MacKenzie, *An Engine, Not a Camera* and Donald MacKenzie, "Long-Term Capital Management and the sociology of arbitrage," *Economy and Society* 32, no. 3 (2003): 349-380.

incorporation of heterogeneity, generativity, and heterarchical search in the Metropolis possibility space of financial analysis software, in concert with the wide variety of other devices, offers to make visible an elusive connectedness within markets, a connectedness which might be ultimately profitable. This visibility emerges from the proliferation of new time series objects, the metrics that transform and compare them, and related visualizations, all of which engage the analyst in continuing cycles of revision and specification. Metropolis engages its users in patterns of dissection, suture, and testing in the search for linkages, a cybernetic paradigm of financial investigation that seeks to understand the "raw" data of financial markets, packaged as time series objects, as terminal inputs and outputs. And when the products of these analyses become the stuff of trading activity, such linkages may indeed take on a certain reality. Do programmers make markets? This question is less clear now, but its seems they do contribute to the distribution of new possibilities that interact with and modify microstructures and create new virtualities and potentialities at scale.

6. Conclusion

The presumed coherence, vitality, but impersonality and continued opacity of markets is part of what grounds inquiry in modern economic discourse and finance theory. This understanding is a relatively recent historical phenomenon, as Timothy Mitchell argues in his critique of the role of development economics in modern Egypt. Mitchell points to the very recent history of "the economy" and "the market" as concepts refer to "a self-contained space, distinct from other social spheres such as the household, the state, or the sphere of culture."[70] Mitchell argues that "[i]n Anglo-American political discourse the market came to stand for a system of forces that the state claimed was independent of its management of the economy, setting limits that this management could not profitably transgress."[71] The result is the discursive construction of a supposedly inscrutable and apolitical realm that nonetheless must be constantly worked on and "fixed" by various state and non-state actors.[72]

Interpretive market devices are not innocent of this more properly political edge to the performativity of markets. The different incorporations of the Black-Scholes-Merton model MacKenzie examines contributed to the production of interlinked global markets, providing justifications for arbitrage practices that bolstered the linkage between stock and futures markets. At the same time, this linkage fostered an unrecognized complexity and precarity, dependent as it was not just on the abstract model but other material infrastructure. The benefits of increasingly deregulated and globalized financial markets are generally enjoyed by a narrow class of investors, but, as has been evidenced on numerous occasions (including the 2008 subprime mortgage crash), it is often states and publics that bear the costs when the consequences of unrecognized interdependence turn into crisis. Metropolis, as a system centered on the recognition of previously unseen market linkages, would seem to be a participant in the wider performances that produce "the global market."

The political consequences of the production of global markets are amplified by the legitimatory effects of economic models and their incorporations. While an economist's model

---

[70] Timothy Mitchell, *Rule of Experts: Egypt, Techno-Politics, Modernity* (Berkeley: University of California Press, 2002), 246.
[71] Ibid.
[72] Timothy Mitchell, "Fixing the Economy," *Cultural Studies* 12, no. 1 (1998): 82-101.

contributes to actors' and society's sense of the rationality of financial markets and exchange behavior, a platform like Metropolis might complement this with a sense of their technicity. When watching a trader or analyst use software like Metropolis, investment appears as an exercise in technical expertise and intelligence distinct from more "vicious" renderings of financial practice as akin to gambling. The drive and sophistication of financial analysis are framed by the presumption that "the market" is an opaque and difficult object of understanding. For traders, the "market" (and only the market) "holds absolute truths," and it acquires a sublimity in the discourse of finance professionals in its continual resistance to their efforts at full understanding.[73] Caitlin Zaloom, in her account of futures traders at in Chicago and London, describes the almost spiritual discipline of traders and the practices of detachment—from personal lives, from pet theories, from past successes and failures—that allow traders to enter "the zone" and "experience the market and become a part of this living thing, intimately connected to it."[74] Similarly, the image of the investment class that emerges from Metropolis demonstrations is not one of plutocracy or excess, but a disciplined cadre of researchers, analysts, nerds, and quants. A key claim legitimized by these technical and intellectual performances is that "the market doesn't care what you think or who you are."[75] These are smart people trying to understand a complex technical domain, and this image of technicity contributes to a sense that markets are distinct and apolitical objects of analysis. The legitimacy of finance becomes entangled with the discourses of meritocracy that circulate in Silicon Valley, and what would once appear as unchecked enthusiasm for profit and accumulation is now understood as a kind of intellectual discipline more closely related to the innocent joys of gaming and puzzle solving.

Finally, Metropolis suggests a shift in the requirements for participation in financial markets. The assemblages entailed by Metropolis deployments—voluminous data, large scale computational infrastructure, and a blend of both technical and economic expertise—are significant barriers to entry and entrench financial markets as the province of select actors. Some of these actors work as proxies for other public or private institutions and individuals, but the shifting, complex, and expensive mechanisms that make markets and knowledge about markets may engender a gap between the financial practice and the practices of accountability that surround it. Images of technical wizardry may be deployed to hide the presence of a deliberate Ponzi scheme, but important issues regarding accountability and the opacity of financial practice are not necessarily criminal in nature. Harry Kat, a Dutch economist investigating claims about hedge fund performance, produced a computer model that could approximate the returns of these funds, raising questions about the effectiveness of proprietary models, "active management," and the fees investors pay for these.[76] The use of technology and expertise is not necessarily related to its performance, even in this most discerning of domains; it is often a function of enthusiasm and habit. In the wake of such enthusiasm, material and social investments create a kind of inertia and gravitational pull with respect to these technologies. In the case of systems like Metropolis, we might ask after the long-term effects of a continued devotion to databased market knowledge and the continuous search for (or construction of) linkages among far-flung sectors and markets.

---

[73] Zaloom, *Out of the Pits*, 127.
[74] Ibid., 128.
[75] Ibid.
[76] John Cassidy, "Hedge Clipping," *The New Yorker*, July 2, 2007, http://www.newyorker.com/magazine/2007/07/02/hedge-clipping.

Epilogue

Palantir's two platforms, Gotham and Metropolis, provide two contemporary exemplars of the entanglements between domains of computation, epistemology, ontology, and politics. On the one hand, Palantir's products are an instructive counterpoint to the current enthusiasm for "Big Data," both in industry and in criticism, that fixates on neural networks and other machine learning systems, portraying a future where apparently monolithic (and still anthropomorphic) systems are the site of intelligent performances. This examination of Palantir has highlighted how intelligence can emerge not from the efficiency of any one system but from the contact and interaction among heterogeneous systems. These are instances of distributed and heterogeneous cognition, "crazy ballet," and disorderly processes that address a world where most interesting problems are, and may likely remain, ill structured.

Palantir's systems, however, share a requirement with other machine-dominated approaches in the contemporary data analysis landscape, the requirement for massive amounts of relational and transactional data in order to make sense of social phenomena. In other words, organizations' investment in these data analysis systems generates a concomitant demand for expanding systems of data collection to feed into them. The sunk costs of such systems for collection and analysis creates the temptation to continually refer epistemic processes to the capabilities of these systems. This temptation recalls Brian Cantwell Smith's concept of the inscription error, that is:

> a tendency for a theorist or observer, first, to write or project or impose or inscribe a set of ontological assumptions onto a computational system (onto the system itself, onto the task domain, onto the relation between the two, and so forth), and then, second to read those assumptions or their consequences back off the system, *as if that constituted an independent empirical discovery or theoretical result*.[1]

Put otherwise, the computational ontologies we develop, especially those for the administration and governance of economies and societies, begin to appear not as tools but as fundamental realities. Combined with the social and material inertia of such systems, the abstractions we have previously projected onto data systems for the purpose of governance may, in the future, come to govern us. This image is not one of human societies taken over by intelligent machines. Rather, it should call our attention to the ways in we may lose sight of and naturalize the ontological commitments in our information technologies. As critical segments of global society become articulated with digital systems, the potential for reciprocal formatting of organizational imperatives by those technologies increases, and the effects of this formatting on the possibilities for perception, knowledge, action, and legitimacy should not be overlooked.

---

[1] Smith, *On the Origin of Objects*, 50.

Palantir is a relatively young and growing technology company with confidential and classified contracts and facing wide public scrutiny, and, as one might expect, the company has removed and changed number of pages on its website in recent years. Two public deployments of their platform, AnalyzeThe.US and Joyride, have also been removed from service. Fortunately, most all of these removed or modified pages have been indexed and archived by projects like the Internet Archive (also known as the Internet Wayback Machine), and in such cases I have included those links where available.

Patents:

Aymeloglu, Andrew, Kevin Simler, Garry Tan, and Derek Cicerone. "Object-oriented time series generator." U.S. Patent 8,484,115, issued July 9, 2013.

Aymeloglu, Andrew, Kevin Simler, Eric Poirier, Garry Tan, and Brandon Burr. "Object modeling for exploring large data sets." U.S. Patent Application 14/618,213, filed February 10, 2015.

Jain, Akash, Robert J. McGrew, and Nathan Gettings. "Creating data in a datastore using a dynamic ontology." US Patent 7,962,495. Filed Nov. 20, 2006, issued June 14, 2011.

Stowe, Geoffrey, Chris Fischer, Paul George, Eli Bingham, and Rosco Hill. "System and method for investigating large amounts of data." US Patent 8,799,240. Filed June 23, 2011, issued August 5, 2014.

Videos:

"Building a Human Trafficking Case from a Lead to Arrests." YouTube video, 20:20. Posted by "Palantir," July 15, 2013. https://www.youtube.com/watch?v=gS1IMB--3dw.

"Dynamic Ontology," YouTube video, 10:30. Posted by "Palantir," July 5, 2012, https://www.youtube.com/watch?v=ts0JV4B36Xw.

"Finding a Mole: Cyber Counter Intelligence – Introduction," YouTube video, 1:43, posted by "Palantir," July 5, 2012. https://www.youtube.com/watch?v=NanaHD7iMLQ.

"Finding a Mole: Cyber Counter Intelligence – Part I," YouTube video, 3:53. Posted by "Palantir," July 5, 2012. https://www.youtube.com/watch?v=53NVooaR4mk.

"Finding a Mole: Cyber Counter Intelligence – Part 2," YouTube video, 4:02. Posted by "Palantir," July 5, 2012. https://www.youtube.com/watch?v=nzX1qRf80yc.

"Finding a Mole: Cyber Counter Intelligence – Part 3," YouTube video, 2:53. Posted by "Palantir," July 5, 2012. https://www.youtube.com/watch?v=Vn1a-hoBSn0.

"Friction in the Machine: How Fluid Processes Allow Optimal Human-Computer Interaction." YouTube video, 38:37. Posted by "Palantir," November 4, 2011. https://www.youtube.com/watch?v=Kw1hZkfOVhQ.

"GovCon7: Introduction to Palantir," YouTube video, 40:01. Posted by "Palantir," November 2, 2011. https://www.youtube.com/watch?v=f86VKjFSMJE.

"Human-Computer Symbiosis: Efficient and Powerful Use of Computing Power." YouTube video, 37:43. Posted by "Palantir," November 3, 2010. https://www.youtube.com/watch?v=oLalkcMDCwg

"Humans, Data, and the Culture of Too Much Information." YouTube video, 23:37. Posted by "Palantir," June 4, 2013. https://www.youtube.com/watch?v=P2tsS4ktt30;

"Interview: Alex Karp, Founder and CEO of Palantir," interview by Evelyn Rusli, YouTube video, 6:40, posted by "TechCrunch," February 16, 2012, https://www.youtube.com/watch?v=VJFk8oGTEs4.

"Multi Level Security – Part 1." YouTube video, 5:06. Posted by "Palantir," July 5, 2012. https://www.youtube.com/watch?v=rmppUPvc-tk.

"Multi Level Security – Part 2." YouTube video, 11:54. Posted by "Palantir," July 5, 2012. https://www.youtube.com/watch?v=wNHfO4Xa9K4.

"Object Model," YouTube video, 15:02. Posted by "Palantir," July 5, 2012. https://www.youtube.com/watch?v=qNxk-1_r7A4.

Palantir Blog posts:

Gesher, Ari. "AtlasDB: Transactions for Distributed Key-Value Stores (Part I)." *The Palantir Blog*. https://palantir.com/2014/06/atlasdb-transactions-for-distributed-key-value-stores-part-i. Page removed, archived version: https://web.archive.org/web/20160403175331/http://www.palantir.com/2014/06/atlasdb-transactions-for-distributed-key-value-stores-part-i/

Gesher, Ari. "Friction in Human-Computer Symbiosis." *The Palantir Blog*. March 8, 2010. Accessed May 9, 2016, https://palantir.com/2010/03/friction-in-human-computer-symbiosis-kasparov-on-chess. Page removed, archived version: https://web.archive.org/web/20160702013739/https://www.palantir.com/2010/03/friction-in-human-computer-symbiosis-kasparov-on-chess/.

Gesher, Ari. "Palantir: embodying a 50-year-old vision of the future?" *The Palantir Blog*, March 16, 2007. Accessed May 9, 2016, https://palantir.com/2007/03/human-computer-symbiosis. Page removed, archived version: https://web.archive.org/web/20140322235929/https://palantir.com/2007/03/human-computer-symbiosis.

Gesher, Ari. "Palantir: like an operating system for data analysis." *The Palantir Blog*, November 6, 2009. http://palantir.com/2009/11/palantir-like-an-operating-system-for-data-analysis. Page removed, archived version: https://web.archive.org/web/20160825230157/http://www.palantir.com:80/2009/11/palantir-like-an-operating-system-for-data-analysis/

Grant, John. "Datafication and You." *The Palantir Blog*, October 25, 2013. Accessed May 9, 2016. https://palantir.com/2013/10/datafication-and-you. Page removed, archived version: https://web.archive.org/web/20150918164903/https://www.palantir.com/2013/10/datafication-and-you/

Simler, Kevin. "Palantir: so what is it you guys do?" *The Palantir Blog*, December 4, 2007. Accessed May 9, 2016. https://palantir.com/2007/12/what-do-we-do. Page removed, archived version: https://web.archive.org/web/20160414140201/https://www.palantir.com/2007/12/what-do-we-do/

Agre, Philip E. *Computation and human experience*. Cambridge, UK: Cambridge University Press, 1997.

Anfinsen, Christian B. "Principles that Govern the Folding of Protein Chains." *Science* 181, no. 4096 (1973): 223-230.

Arquilla, John and David F. Ronfeldt. *Networks and Netwars: The Future of Terror, Crime, and Militancy*. Santa Monica: RAND, 2001.

Austin, J.L. *How to Do Things With Words*. Cambridge, MA: Harvard University Press, 1962.

Barad, Karen. "Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter." *Signs* 28 no. 3 (Spring 2003), 801-831.

Berry, David M. "A Contribution Toward a Grammar of Code." *The Fibreculture Journal* 13 (2008). Accessed July 17, 2017. http://thirteen.fibreculturejournal.org/fcj-086-a-contribution-towards-a-grammar-of-code/.

Beunza, Daniel and Raghu Garud. "Calculators, lemmings, or frame-makers? The intermediary role of securities analysts." In *Market Devices*, eds. Fabian Muniesa, Yuval Millo, and Michel Callon. Oxford: Blackwell, 2007.

Beunza, Daniel and David Stark. "How to Recognize Opportunities: Heterarchical Search in a Trading Room." In *The Sociology of Financial Markets*, eds. Karin Knorr Cetina and Alex Preda. Oxford University Press, 2005.

Biggerstaff, Ted J. "Design Recovery for Maintenance and Reuse." *Computer* 22, no.7 (1989): 36-49.

Boecking, Benedikt, Margeret Hall, and Jeff Schneider. "Event Prediction With Learning Algorithms—A Study of Events Surrounding the Egyptian Revolution of 2011 on the Basis of Micro Blog Data." *Policy & Internet* 7, no. 2 (2015): 159-184.

Bogost, Ian. "The Rhetoric of Video Games." In *The Ecology of Games: Connecting Youth, Games, and Learning*, ed. Katie Salen. Cambridge, MA: The MIT Press, 2008.

Borgida, Alexander, Marco A. Casanova, and Alberto H.F. Laender. "Logical Database Design." In *Encyclopedia of Database Systems*, eds. Ling Liu and M. Tamer Özsu. New York: Springer 2009.

Bourdieu, Pierre. *The Social Structures of the Economy*. Cambridge, UK: Polity Press, 2005.

Brandes, Ulrik, Linton C. Freeman, and Dorothea Wagner. “Social Networks.” In *Handbook of graph drawing and visualization*, ed. Roberto Tamassia. CRC Press, 2013.

Brandes, Ulrik, Patrick Kenis, and Jörg Raab. “Explanation Through Network Visualization.” *Methodology* 2, no.1 (2006): 16-23.

Butler, Judith. “Performative Agency.” *Journal of Cultural Economy* 3, no. 2 (2010): 147-161.

Callon, Michel. “Some elements of a sociology of translation: domestication of the scallops and the fishermen of St Brieuc Bay.” *The Sociological Review* 32, no. 1 (1984): 196-233.

———. “An essay on enframing and overflowing: economic externalities revisited by sociology,” in *The Laws of the Markets*, ed. Michel Callon. Oxford: Blackwell, 1998.

———. “Introduction: the embeddedness of economic markets in economics.” In *The Laws of the Markets*, ed. Michel Callon. Oxford: Blackwell, 1998.

Smith, Brian Cantwell. *On the Origin of Objects*. Cambridge, MA: MIT Press, 1996.

Cassidy, John. “Hedge Clipping.” *The New Yorker*, July 2, 2007. http://www.newyorker.com/magazine/2007/07/02/hedge-clipping.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey.” *ACM computing surveys (CSUR)* 41, no. 3 (2009): 15.

Chatman, Seymour. *Story and Discourse: Narrative Structure in Fiction and Film*. Ithaca: Cornell University Press, 1979.

Cheney-Lippold, John. “A New Algorithmic Identity: Soft Biopolitics and the Modulation of Control.” *Theory, Culture & Society* 28, no. 6 (2011): 164-181.

Chikofsky, E.J., and J.H. Cross. “Reverse engineering and design recovery: A taxonomy.” *IEEE Software* 7, no. 1 (1990): 13-17.

Chun, Wendy H.K. “Crisis, crisis, crisis, or sovereignty and networks.” *Theory, Culture & Society* 28, no. 6 (2011): 91-112.

Cole, David. “’We Kill People Based on Metadata.’” NYRDaily Blog, May 10, 2014, *The New York Review of Books*. http://www.nybooks.com/daily/2014/05/10/we-kill-people-based-metadata/.

Collier, Stephen. “Topologies of Power: Foucault’s Analysis of Political Government beyond ‘Governmentality’.” *Theory, Culture & Society* 26, no. 6 (2009): 78-108.

Cooper, Seth, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popovic, and Foldit players. "Predicting protein structures with a multiplayer online game." *Nature* 466 (2010): 756-760.

Das, Rhiju and David Baker. "Macromolecular Modeling with Rosetta." *Annual Review of Biochemistry* 77 (2008): 363-82.

DeCandia, Giuseppe, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels. "Dynamo: Amazon's Highly Available Key-value Store." *ACM SIGOPS Operating Systems Review* 41, no. 6 (2007): 205-220.

Delevett, Peter. "Palantir Technologies' latest funding round could top $200 million." *San Jose Mercury*, September 27, 2013. http://www.mercurynews.com/2013/09/27/exclusive-palantir-technologies-latest-funding-round-could-top-200-million/.

Demers, Alan, Dan Greene, Carl Huster, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dane Swinehart, and Doug Terry. "Epidemic Algorithms for Replicated Database Maintenance." *ACM SIGOPS Operating Systems Review* 22, no. 1 (1988): 8-32

DeWitt, David, and John Gray. "Parallel Database Systems: The Future of High Performance Database Systems." *Communications of the ACM* 35, no. 6 (1992), 85-98.

Engelbart, Douglas. *Augmenting Human Intellect: A Conceptual Framework*. Menlo Park: Stanford Research Institute, 1962.

Ernst, Wolfgang. "Media Archaeography." In *Media Archaeology: Approaches, Applications, and Implications.* Edited by Erkki Huhtamo and Jussi Parikka. Berkeley: University of California Press, 2011.

Fetzer, J.K. *Artificial Intelligence: Its Scope and Limits*. Dordrecht: Kluwer Academic, 1990.

Forrest, Stephanie, Steven A. Hofmeyr, and Anil Somayaji. "Computer immunology." *Communications of the ACM* 40, no. 10 (1997): 88-96.

Forrest, Stephanie, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri. "Self-nonself discrimination in a computer." *Proceedings of 1994 IEEE Symposium on Research in Security and Privacy*: 202-212.

Foster, Robert. *The Complete Guide to Middle-Earth: From the Hobbit through The Lord of the Rings and beyond*. New York: Ballantine Books, 2001.

Foucault, Michel. *Security, Territory, Population*. Edited by Michel Senellart. Translated by Graham Burchell. New York: Palgrave Macmillian, 2007.

Freeman, Linton. “Visualizing Social Networks.” *Journal of Social Structure* 1, no. 1 (2000): 4. Retrieved from http://www.cmu.edu/joss/content/articles/volume1/Freeman.html.

Girard, Monique and David Stark. “Distributing intelligence and organizing diversity in new-media projects.” *Environment and Planning A* 34 (2002): 1927-1949.

Granovetter, Mark. “The Strength of Weak Ties.” *American Journal of Sociology* 78, no. 6 (May, 1973): 1360-1380.

Granovetter, Mark. “The Impact of Social Structure on Economic Outcomes.” *The Journal of Economic Perspectives* 19 (2005): 33-50.

Gruber, Tom. “Ontology.” In *Encyclopedia of Database Systems*. Edited by Ling Liu and M. Tamer Özsu. New York: Springer 2009.

Halliday, Josh. “Guardian Activate 2010: live coverage.” *The Guardian*, July 1, 2010. https://www.theguardian.com/media/pda/2010/jun/30/guardian-activate-summit-2010-liveblog.

Halpern, Orit. *Beautiful Data: A History of Vision and Reason since 1945*. Duke University Press, 2014.

Haraway, Donna. *Simians, Cyborgs, and Women: The Reinvention of Nature.* New York: Routledge, 1991.

Hardy, Quentin. “Unlocking Secrets, if Not Its Own Value.” *The New York Times*, June 1, 2014. http://www.nytimes.com/2014/06/01/business/unlocking-secrets-if-not-its-own-value.html.

Hayashibara, Naohiro, Xavier Defago, Rami Yared, and Takuya Katayama. “The φ accrual failure detector.” In *Proceedings of the 23rd IEEE International Symposium on Reliable Distributed Systems, 2004,* 66-78.

Hayles, N. Katherine. “Narrative and Database: Natural Symbionts.” *PMLA* 122 no. 5 (Oct. 2007): 1603-1608.

Hayles, N. Katherine. *How We Became Posthuman*. Chicago: UChicago Press, 1999.

Holm, Peter. “Which Way Is Up on Callon?” In *Do Economists Make Markets?: On the Performativity of Economics*. Edited by Donald MacKenzie, Fabian Muniesa, and Lucia Siu. Princeton University Press, 2007.

Hu, Tung-Hui. “Real time/zero time.” *Discourse* 34, no. 2 (2012): 163-184.

Jannotti, John, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole, Jr. "Overcast: reliable multicasting on an overlay network." In *Proceedings of the 4th conference on Symposium on Operating System Design & Implementation* (2000).

Kasparov, Garry. "The Chess Master and the Computer." *The New York Review of Books*, February 11, 2010. http://www.nybooks.com/articles/2010/02/11/the-chess-master-and-the-computer/.

Kent, William. *Data and Reality*. 2000.

Klovdahl, Alden. "A Note on Images of Networks." *Social Networks* 3 (1981): 197-214.

Knorr Cetina, Karin. *Epistemic Cultures: How the Sciences Make Knowledge*. Cambridge: Harvard University Press, 1999.

Knorr Cetina, Karin and Urs Bruegger. "Global Microstructures: The Virtual Societies of Financial Markets." *American Journal of Sociology* 107, no. 4 (2002): 905-950.

Knorr Cetina, Karin and Barbara Grimpe. "Global Financial Technologies: Scoping Systems That Raise the World." In *Living in a Material World: Economic Sociology Meets Science and Technology Studies*. Edited by Trevor Pinch and Richard Swedberg. Cambridge, MA: MIT Press, 2008.

Lakshman, Avinash, and Prashant Malik. "Cassandra - A Decentralized Structured Storage System." *ACM SIGOPS Operating Systems Review* 44, no. 2 (2010): 35-40.

Latour, Bruno. "Visualization and Cognition: Drawing Things Together." In *Knowledge and Society: Studies in the Sociology of Culture Past and Present*. Edited by H. Kuklick. Jai Press, 1986.

Latour, Bruno, and Steve Woolgar. *Laboratory Life: The Construction of Scientific Facts*. Beverly Hills: Sage, 1979.

Licklider, J.C.R. "Man-Computer Symbiosis." *IRE Transactions on Human Factors in Electronics* 1 (1960): 4-11.

Lohr, Steve. "A Data Weapon to Avoid the Next Financial Crisis." *The New York Times Bits Blog*, September 9, 2013. https://bits.blogs.nytimes.com/2013/09/09/a-data-weapon-to-avoid-the-next-financial-crisis/.

Mackenzie, Adrian, and Ruth McNally. "Living Multiples: How Large-scale Scientific Data-mining Pursues Identity and Difference." *Theory, Culture & Society* 30, no. 4 (2013): 73.

MacKenzie, Donald. "Long-Term Capital Management and the sociology of arbitrage." *Economy and Society* 32, no. 3 (2003): 349-380

———. *An Engine, Not a Camera*. Cambridge, MA: MIT Press, 2006.

Magnani, Lorenzo. *Abductive Cognition: The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning*. Berlin: Springer, 2009.

Maus, Gregory. "A (Pretty) Complete History of Palantir." August 11, 2015. http://www.socialcalculations.com/2015/08/a-pretty-complete-history-of-palantir.html

Timothy Mitchell, "Fixing the Economy," *Cultural Studies* 12, no. 1 (1998): 82-101.

———. *Rule of Experts: Egypt, Techno-Politics, Modernity*. Berkeley: University of California Press, 2002.

Mohan, C. "History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla." In *Proceedings of the 16th International Conference on Extending Database Technology* (2013): 11-16.

Moreno, Jacob. *Who Shall Survive? Foundations of Sociometry, Group Psycho-therapy, and Sociodrama*. Beacon, NY: Beacon House, 1953.

Myers, Natasha. *Rendering Life Molecular: Models, Modelers, and Excitable Matter*. Duke University Press, 2015.

"PAL / CSS report from the dark horse's mouth." *ChessBase*, June 22, 2005. http://en.chessbase.com/post/pal-c-report-from-the-dark-horse-s-mouth.

Parrika, Jussi. "Operative Media Archaeology: Wolfgang Ernst's Materialist Media Diagrammatics." *Theory, Culture & Society* 28, no. 5 (2011): 52-74.

Patterson, David A. "The Data Center is the Computer." *Communications of the ACM* 51, no. 1 (January 2008): 105.

Pavlo, Andrew, "A Comparison of Approaches to Large-Scale Data Analysis" (2009).

Rekoff, Michael G. "On reverse engineering." *IEEE Transactions on systems, man, and cybernetics* 2 (1985): 244-252.

Rowstron, Antony, and Peter Druschel. "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems." In *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pp. 329-350. Springer, Berlin, Heidelberg, 2001.

Schneier, Bruce. "NSA Surveillance: What We Know, and What to Do about It." YouTube video, 53:39. Posted by "RSA Conference," May 5, 2014. https://www.youtube.com/watch?v=1iMFPMqboZc.

Simon, Herbert. “The Structure of Ill Structured Problems.” *Artificial Intelligence* 4 (1973): 181-201.

———. *The Sciences of the Artificial* (Cambridge, Mass.: MIT Press, 1996)

Sockut, Gary H., and Robert P. Goldberg. “Database Reorganization – Principles and Practice.” *ACM Computing Surveys* 11 no. 4 (Dec. 1979), 371-395

Somerville, Heather. “Uber valued at whopping 18.2 billion.” *San Jose Mercury News*, June 6, 2014. http://www.mercurynews.com/business/ci_25913159/18-2-billion-valuation-makes-uber-top-venture.

Spinellis, Diomidis. “Another Level of Indirection.” In *Beautiful Code: Leading Programmers Explain How They Think*. Edited by Andy Oram and Greg Wilson. Sebastopol, CA: O’Reilly, 2007.

Stoica, Ion, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. “Chord: A scalable peer-to-peer lookup service for internet applications.” *ACM SIGCOMM Computer Communication Review* 31, no. 4 (2001): 149-160.

Stonebraker, Michael. “The Case for Shared Nothing.” *IEEE Database Eng. Bull.* 9, no. 1 (1986): 4-9.

Suchman, Lucy. “Situational Awareness: Deadly Bioconvergence at the Boundaries of Bodies and Machines.” *MediaTropes* 5, no. 1 (2015): 1-24.

Thompson, Charis. *Making Parents: The Ontological Choreography of Reproductive Technologies*. Cambridge: The MIT Press, 2005.

Wang, Wego. *Reverse Engineering: Technology of Reinvention*. Boca Raton: CRC Press, 2011.

Weber, Samuel. “Targets of Opportunity: Networks, Netwar, and Narratives.” *Grey Room* 15 (2004): 6-27.

Woodman, Spencer. “Palantir Provides the Engine for Donald Trump’s Deportation Machine.” *The Intercept*, March 2, 2017. https://theintercept.com/2017/03/02/palantir-provides-the-engine-for-donald-trumps-deportation-machine/.

Zaleski, Oliva. “WeWork Becomes World’s Fifth Most-Valuable Startup.” *Bloomberg Technology*, July 12, 2017. https://www.bloomberg.com/news/articles/2017-07-12/wework-is-said-to-become-world-s-fifth-most-valuable-startup.

Zaloom, Caitlin. *Out of the Pits: Traders and Technology from Chicago to London*. University of Chicago Press, 2006.