

UCSF

UC San Francisco Electronic Theses and Dissertations

Title

Model-based interpretation of time-ordered medical data

Permalink

<https://escholarship.org/uc/item/1755k2mz>

Author

Kahn, Michael G.

Publication Date

1989

Peer reviewed|Thesis/dissertation

Model-Based Interpretation of Time-Ordered Medical Data

by

Michael Gary Kahn

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Medical Information Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA

San Francisco

Date

University Librarian

Degree Conferred: JAN 2 1989

© Copyright 1989
by
Michael Gary Kahn

Acknowledgments

I often have wondered why wives, kids, pets, and parents always seem to get mentioned last in most acknowledgments. Everybody knows that, without these stalwart bodies rooting for you through the dark days and tolerating you through the exuberant days, the acknowledgment page would not even exist. So, here is my contribution to a new trend: I want to thank my wife, Mary, and my son, Daniel, for always being there.

Most medical centers have no knowledge about medical informatics, and even fewer centers know what to do with a person who wants to study in this field. Despite this uncertain status, Wallace V. Epstein M.D., director of the Robert Wood Johnson Clinical Scholars Program at UCSF, gave me the opportunity to experience real medical informatics research at a time when I was uncertain whether the field would be as exciting when I was an active participant as it seemed to be when I was a passive observer. Dr. Epstein's willingness to offer me a highly desired RWJ Clinical Scholars Fellowship gave me the chance to find my life-long career. I am tremendously grateful to him and to the Johnson Foundation for their courage and support.

All research projects have periods of ups and downs. Two other people stuck by my side through the good and through the bad. My eternal thanks and gratitude to both Larry Fagan and Lewis Sheiner for their commitment to my professional development during this research.

Because of my interests in temporal reasoning, I eventually moved my research base to the Medical Computer Sciences Group at Stanford University, headed by Edward Shortliffe. At UCSF, the Section on Medical Information Sciences was headed by Scott

Blois. My sincere thanks to both Drs. Shortliffe and Blois for never letting geographical or administrative technicalities prevent me from pursuing my research. Especially remarkable was the willingness of the Stanford group to provide me with any and all needed resources, the most precious resource being their personal time and attention. On July 15, 1988, Dr. Blois passed away. He did not get to read the final draft of this dissertation; his comments and insight are deeply missed.

It is a fact that the administrative secretaries keep the ship afloat. The UCSF and Stanford ships are no exceptions. For not getting bent out of shape because of my odd interinstitutional status, my thanks to Marina Gordillo, at UCSF, and Darlene Vian, at Stanford. They were always calm when I was frazzled but, then again, den mothers always are.

My thanks to Mark Musen and Curt Langlotz for all the special dinners and frank talks about medical informatics, academic politics, and the world around us. Both the “outhouse” and the MSOB “senior student” corner will be sorely missed.

During the 5 years of commuting between San Francisco and Stanford, I logged approximately 85,000 miles on Highway I-280. My final thanks go to 1EPR625, a 1982 Honda Accord that never broke down, and to the statue of Father Junipero Serra, who always got me there and back safely.

Preface

This dissertation is about temporal reasoning and temporal representation. These concepts are interpreted in different ways by different people. Because this work deals with a range of temporal issues, the meanings of the terms *temporal reasoning* and *temporal representation* change as the research focus changes. Over the course of this dissertation, the following variations on the role of time in problem solving are examined:

- Storing and retrieving time-varying, context-sensitive data
- Detecting temporal relationships from time-ordered data
- Reasoning about change and processes
- Representing dynamic systems
- Reasoning about states and contexts

Providing a single definition of *temporal reasoning* and *temporal representation* that captures the whole breadth of these issues would be as uninformative as saying “This dissertation is about time.” Instead, I will state which emphasis is in the current spotlight as I move from one perspective to another.

This work draws from many research areas, such as *database management*, *mathematical modeling*, *symbolic knowledge representation*, and *temporal inferencing*. Depending on the investigator’s discipline, similar terms have different meanings. A worrisome example is *temporal database*. This term is used frequently within both the database and

artificial-intelligence communities, but it refers to markedly different systems that have little in common except for a special concern about time. The term *temporal constraints* has a vastly different interpretations in the artificial-intelligence and in the mathematics communities. These terminological differences interfere with comparing ideas from one field to those of the other. This issue becomes most apparent in Chapter 2, where I bring together common pieces from many places. A useful mechanism for enforcing a semblance of order on this chaos is to adopt a set of conventions at the outset and to stick to them for the duration of the work. I use masculine pronouns for all third-person references. I use upper case type for the names of programs and data structures, and italics for variable names and variable values.

The potential audience for this work is diverse. The issues that I touch on are of interest to physicians, system modelers, computer scientists, and builders of decision-support systems. In imagining the reader for whom I was writing, I made some assumptions about his knowledge of medicine and computers. I have not defined all the medical and the computer-science terms; an absolute novice in one of the fields cannot be educated by reading this dissertation. For the nonphysician, the medical issues are not too technical, provided the reader has some understanding of medical problem solving. My work can be understood, but cannot be fully appreciated, by a reader who has never experienced the frustration of trying to interpret the mess that constitutes the typical patient chart. My emphasis, however, is not on medicine; rather, it is on the technical insights of interest to the builder of decision support-systems that must handle time-ordered data. Perhaps the best description of my envisioned reader is someone interested in an old field with a new name—*medical informatics*. This discipline brings together people with both technical and clinical skills who seek computer-based solutions to medical information problems. This dissertation is about patient-specific medical information—its organization, interpretation, and use in decision making. These concerns are central in medical informatics.

The remainder of this preface provides a succinct overview of each of the chapters. The reader can glean the key points of this dissertation by reading this section, Chapter 1, and Chapter 9. The technical details are presented in Chapters 4 through 8. Chapters 2 and 3 provide the background material that gives perspective to the new approaches presented in the dissertation. A number of closely related research projects and programs are central to this research. In the dissertation, I discuss two versions of an expert system for cancer chemotherapy advise (mainframe-ONCOCIN and workstation-ONCOCIN), two

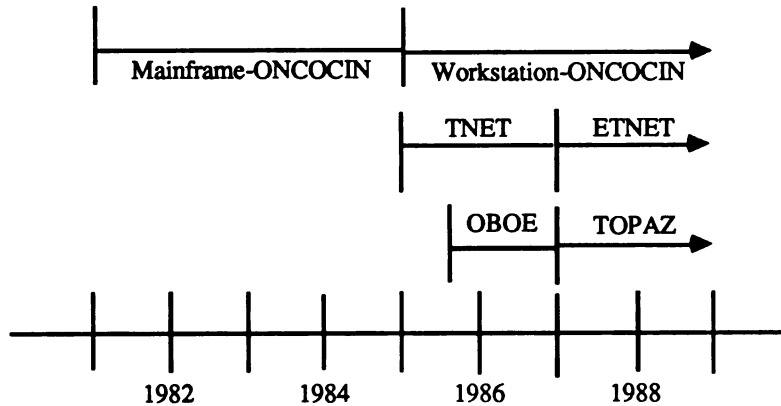


Figure 0.1 Temporal sequence of relevant programs.

The research described in this dissertation focuses on two implementations of an expert system for cancer chemotherapy (mainframe-ONCOCIN and workstation-ONCOCIN), two versions of a temporal database (TNET and ETNET), and a program that summarizes patient records (TOPAZ). This diagram shows the temporal relationships among these programs.

versions of a temporal database (TNET and ETNET), and a program for summarizing time-ordered clinical data (TOPAZ). Figure 0.1 shows the temporal relationships among these programs. Chapters 4 through 8 present the research in this general sequence.

In Chapter 1, *Introduction*, I present the fundamental research issues that motivate this work. I discuss the problems of representing temporal relationships and of reasoning about change in intelligent computer systems. I examine the importance of deriving subtle inferences from time-ordered observations drawn from complex dynamic systems. I argue that, in a setting where changing contexts are common, special methods must be used to limit the knowledge and inferences that are applicable in a specific reasoning context. The goal of this chapter is to provide the necessary problem background and conceptual framework for the remaining chapters.

In Chapter 2, *Temporal Models in Medical Decision Support*, I examine the role of temporal reasoning in medical decision making and in medical decision-support systems. I discuss the issues in modeling the medical environment from the perspective of an underlying medical database that stores observations taken from the medical setting and of a medical expert system that must reason with these observations. I review the problems and

limitations of prior expert systems that had weak or incorrect temporal models. The main theme of this chapter is that no single modeling methodology is rich enough to encode all the temporal relationships that are relevant to medical decision support.

In Chapter 3, *Temporal Reasoning and the Medical Record*, I discuss the diagnostic and therapeutic importance of a patient's medical record. I argue that significant, clinically useful information exists in the temporal patterns among clinical observations. The key problem is that we cannot access these temporal features simply by inspecting single measurements. I review the temporal reasoning skills that must be applied for the successful summarization of a patient's clinical course. This analysis shows that the summarization task must address the same temporal issues as those first raised in Chapter 2. In particular, the summarization task embodies temporal issues related to both databases and knowledge bases. The chapter closes with an examination of the techniques and methods explored by previous research projects that reasoned about time-ordered data.

In Chapter 4, *The ONCOCIN Temporal Network*, I introduce the ONCOCIN project, an expert system developed at Stanford University to provide decision support for physicians caring for ambulatory patients receiving experimental cancer treatments. ONCOCIN provided the research environment for exploring temporal-reasoning issues in medical therapy planning. I discuss the ONCOCIN temporal network (TNET), which is the basis of the symbolic temporal models used in TOPAZ (Chapter 7). I show how the TNET functional requirements were influenced by the interplay between the ONCOCIN database and the ONCOCIN expert system. A temporal query language, TQUERY, is presented in detail. TQUERY supports efficient date-free queries of the ONCOCIN patient database by using TNET to navigate context-sensitive query requests. The main point of this chapter is to show how the ONCOCIN TNET is a reasonable and natural solution to the need for rapid access to time-sensitive patient data, given the temporal features of the ONCOCIN medical problem and domain model.

In Chapter 5, *The TOPAZ Methodology*, I propose a new methodology for interpreting time-ordered data. The features of the summarization task lead to the design features of TOPAZ, a prototype system that summarizes computer-based time-ordered patient records obtained from ONCOCIN.

The TOPAZ methodology is a two-pass approach to temporal data interpretation. The first pass converts superficial observations into implications about the system's underlying entities and processes. The second pass converts these low-level system features into

temporal abstractions that have relevance or importance to decision making. The decision-making task or goal generally determines which features or concepts are important. Therefore, the aggregates created by the second pass must be problem- and task-specific. A final step converts the high-level temporal abstractions into textual form for human review.

The key point in the TOPAZ methodology is the use of multiple temporal models; the first temporal model is responsible for mapping observations into plausible implications based on the explicit constraints imposed by the underlying system structure; the second temporal model maps from low-level system implications derived by the first model into high-level abstractions. I argue that it is not possible to map directly from superficial observations to summary conclusions without intermediate model-based reasoning. Without an explicit system model, key knowledge about temporal relationships and constraints contained in the patterns embedded in time-ordered observations cannot be fully exploited.

In Chapters 6 through 8, I present the technical details of a system called TOPAZ that performs summarization of patient records using the methodological principles presented in Chapter 5. TOPAZ contains three temporal models that work cooperatively to produce a summary of time-ordered clinical data. In Chapter 6, *A Process Model of Granulocyte Kinetics and Chemotherapy*, I present the details of a *numeric model* that represents the physiological entities and functional dependencies that constrain the possible interpretations of the temporal changes in clinical observations. In Chapter 7, *An Interval Model of Temporal Abstraction*, I present a *symbolic model* that converts a time-ordered series of patient-specific physiological model predictions into clinically meaningful temporal abstractions. A new symbolic modeling system called ETNET performs the high-level temporal abstraction. In Chapter 8, *A Model of Summary Text Generation*, I present a *text-generation model* that converts patient-specific observations and abstractions into textual summaries that express clinical concepts in the language and style used by physicians.

In Chapter 9, *Summary and Conclusions*, I review the main experimental hypotheses presented in Chapter 1 in light of the results from building TOPAZ. For each hypothesis, I discuss TOPAZ and its relationship to the works presented in previous chapters. The goal of this integration is to highlight the new contributions of TOPAZ in temporal reasoning and representation and to identify areas that are open to future research.

Abstract

MODEL-BASED INTERPRETATION OF TIME-ORDERED MEDICAL DATA

Michael Gary Kahn

University of California, San Francisco

Time-ordered data often contain implications that physicians infer only by recognizing subtle temporal patterns. This dissertation is concerned with representing and using medical knowledge about temporal relationships so that decision-support programs can deduce abstract clinical concepts that appear as temporal patterns in time-ordered clinical observations.

Computational models of time that are based on a single representational method are limited by the expressive power of the chosen method. I propose that combining multiple temporal formalisms facilitates the creation of robust temporal models. The use of multiple modeling techniques permits each method to represent those temporal relationships that are best expressed within a particular formalism.

This dissertation proposes a methodology for interpreting time-ordered clinical data using a combination of numeric and symbolic techniques. The methodology proposes three transformations: (1) model-based transformation of time-ordered observations into underlying concepts, (2) interval-based transformation of dynamic model predictions into domain-specific temporal abstractions, and (3) state-based transformation of temporal

abstractions into clinically meaningful discourse fragments. The temporal model used to convert observations into underlying system (patient) concepts encodes physiological process knowledge. The temporal model used to convert model-based predictions into temporal abstractions encodes domain-specific event knowledge. These two temporal models differ in the temporal structure they assume and in the temporal concepts they encode. Without multiple temporal models, this diversity of temporal knowledge could not be represented.

Effective medical decision making demands that physicians recognize and exploit the complex temporal changes that permeate the medical record. In this dissertation, I examine the temporal features of time-ordered clinical data in the context of summarizing a patient's medical record. I have demonstrated the three-transformation methodology in a computer program, call TOPAZ, that summarizes the clinical course of individual patients who are receiving experimental cancer chemotherapy.

Table of Contents

Acknowledgments	iii
Preface	v
Abstract	x
List of Tables	xvi
List of Figures	xvii
1 Introduction	1
1.1 The Time Problem	5
1.2 Dynamic Systems	8
1.3 Temporal Inferences from Time-Ordered Data	9
1.4 Context-Sensitive Reasoning	11
1.5 Overview of Key Contributions	12
1.5.1 Combining Quantitative and Qualitative Temporal Models	13
1.5.2 Temporal Extensions to the Time-Oriented-Data Model	15
1.5.3 Context-Sensitive Reasoning	18
1.5.4 Summary	19
2 Temporal Models in Medical Decision Support	20
2.1 Computational Models of Time	22
2.1.1 An Interval-Based Model of Time	23
2.1.2 A Mechanism-Based Model of Processes	24
2.1.3 Ontologic Differences	27
2.1.4 Complexity and Computability in Temporal Models	29
2.1.4.1 Representational Complexity	29
2.1.4.2 Inferential Complexity	31

2.2	Computational Models of Temporal Data	33
2.2.1	Temporal Data Models	35
2.2.1.1	Temporal Extensions to Traditional Data Models	36
2.2.1.1.1	Snapshot Databases	37
2.2.1.1.2	Rollback Databases	37
2.2.1.1.3	Historical Databases	38
2.2.1.1.4	Temporal Databases	40
2.2.1.2	Novel Data Models	41
2.2.1.2.1	Network-Based Temporal Databases	41
2.2.1.2.2	Constraint Networks	43
2.2.2	Temporal Query Languages	46
2.2.2.1	HRDM	46
2.2.2.2	TQUEL	47
2.2.2.3	RX	50
2.3	Computational Models of Temporal Decision Making	51
2.3.1	State-Based Temporal Ignorance	53
2.3.2	Causal-Based Temporal Ignorance	54
2.3.3	Discrete Temporal Models	56
2.3.3.1	VM	56
2.3.3.2	TEMPORAL CONTROL STRUCTURE	59
2.3.3.3	Markov Models of Prognosis	60
2.3.4	Continuous Temporal Models	62
2.3.4.1	DIGITALIS THERAPY ADVISOR	62
2.3.4.2	HEART FAILURE PROJECT	63
2.4	Summary	63
3	Temporal Reasoning and the Medical Record	66
3.1	The Problem Domain: Cancer Chemotherapy	67
3.2	The Medical Record	68
3.3	Summarizing Time-Ordered Clinical Data	71
3.3.1	VM	73
3.3.2	RX	74
3.3.3	Downs' Summarization Program	76
3.3.4	IDEFIX	79
3.3.5	Lessons Learned	81
3.4	Summarizing Cancer Chemotherapy	84
3.4.1	OBOE	86
3.4.2	Implications for Summarizing Time-Ordered Medical Data	91
4	The ONCOCIN Temporal Network	92
4.1	ONCOCIN	94
4.2	The ONCOCIN Temporal Model	98
4.3	The Temporal Network	99
4.3.1	Logical Implementation	100
4.3.2	Physical Implementation	104

4.3.2.1	The TNET Space	106
4.3.2.2	The TCHILDREN Space	107
4.3.2.3	The RESULT space	108
4.4	TQUERY: A Temporal Query Language	108
4.5	TNET Limitations	110
4.6	Summary	113
5	The TOPAZ Methodology	114
5.1	Model-Based Data Interpretation	116
5.2	Interval-Based Data Abstraction	122
5.3	Problem-Based Text Generation	127
5.4	Model-Based Summarization	130
5.5	The TOPAZ Implementation	132
6	A Process Model of Granulocyte Kinetics and Chemotherapy	137
6.1	A Numeric Process Model of the Bone Marrow	139
6.2	The Structural Model	145
6.2.1	Model Equations	145
6.2.2	Model Parameters	145
6.2.3	Model States	148
6.2.4	Model Predictions	149
6.3	The Model-Fitting Procedures	149
6.3.1	Least-Squares Estimation	152
6.3.2	Robust Estimation	154
6.3.3	Examples of Model Fitting	156
6.4	Problems and Extensions	157
6.4.1	Model Inadequacies	158
6.4.2	Parameter Assumptions	160
7	An Interval Model of Temporal Abstraction	165
7.1	Interval-Based Context Models	166
7.2	ETNET	167
7.2.1	Context-Specific Reasoning	169
7.2.2	ETNET RULENODES	170
7.2.3	Temporal Pattern-Matching and Searching	173
7.3	The TOPAZ Temporal Model	178
7.4	Problems and Extensions	184
7.4.1	The ETNET system	184
7.4.1.1	The Representation of Temporal Knowledge using ETNET	186
7.4.1.2	Temporal Reasoning	186
7.4.1.3	Database Management	188
7.4.2	Model-Based Temporal Abstraction	190
7.5	Summary	196

8	A State-Based Model of Summary-Text Generation	197
8.1	Augmented Transition Networks for Text Generation	200
8.2	Text-Generation Procedures	202
8.3	Problems and Extensions	205
8.4	Summary	206
9	Summary and Conclusions	208
9.1	Summary of the Dissertation	208
9.2	Limitations of the Methodology	210
9.3	Multiple Temporal Formalisms	212
9.4	Model-Based Data Interpretation	217
9.5	Context-Sensitive Reasoning	221
9.6	Looking Forward	223
9.7	Summary	224
	Bibliography	226
A	TQUERY BNF Syntax	238
B	ETNET Rule Predicates and Actions	240
C	TOPAZ-Generated Patient Summaries	243

List of Tables

2.1	Thirteen possible relations between two temporal intervals.	25
2.2	A taxonomy of traditional data-model extensions.	37
2.3	RX time-dependent database-access functions.	52
4.1	TQUERY examples with increasing level of complexity.	111
5.1	A nomenclature for related patient models.	120
6.1	Relative myelosuppressive potency of MOPP drugs.	142
6.2	Parameters of the TOPAZ bone marrow model.	147
6.3	The TOPAZ bone-marrow model states.	148
7.1	Temporal-searching algorithms.	177
7.2	Pair-wise comparisons for summarization and explanation.	179
7.3	Pair-wise comparisons for summarization using data smoothing.	195
8.1	The TOPAZ organizational schema.	198
B.1	ETNET rule predicates.	241
B.2	ETNET rule actions.	242

List of Figures

0.1	Temporal sequence of relevant programs.	vii
1.1	Portions of a patient record and of a TOPAZ summary.	3
1.2	Two methods for determining a flow rate.	10
1.3	A sample patient flowsheet.	14
1.4	The use of multiple temporal models for temporal data summarization. . .	16
1.5	A graphical representation of a temporal network.	17
2.1	Representation of temporal uncertainty in the interval calculus.	25
2.2	A physical system represented as a qualitative model.	27
2.3	Interval representation of changing facts.	28
2.4	Translation of interval relations into point relations.	32
2.5	Three methods used by the TIME SPECIALIST to organize temporal state- ments.	43
2.6	RANGE RELATIONS in the TEMPORAL UTILITY PACKAGE.	44
2.7	Temporal rule predicate in THRIPHT.	45
2.8	A representative simple TQUEL query.	48
2.9	A complex TQUEL statement with temporal-predicate operators.	49
2.10	Atemporal and temporal INTERNIST-1 patient manifestations.	54
2.11	Formalisms for encoding causal knowledge.	55
2.12	Causal abstraction hierarchy in ABEL.	57
2.13	The state-transition graph from the VM system.	58
2.14	Sample rule from VM.	59
2.15	Partitioning of time into homogeneous intervals by TCS.	60
2.16	A three-state Markov model.	61
2.17	A portion of the cardiovascular-network model in HFP.	64
3.1	A patient's white blood cell count following a single dose of chemotherapy. .	70
3.2	A summarization plot generated by VM.	75
3.3	A sample patient record plotted by RX.	77
3.4	The knowledge-base structure in Downs' summarization program.	78
3.5	Graphical output of Downs' summarization program.	80

3.6	A time-oriented probability function in IDEFIX.	82
3.7	Mechlorethamine-induced changes in a patient's white blood cell count. . .	85
3.8	A sample patient record and OBOE output.	87
3.9	Two-pass system design in OBOE.	89
3.10	OBOE augmented transition network encoding of treatment states.	90
4.1	Sample translated rule from ONCOCIN.	96
4.2	LISP predicates in the two ONCOCIN prototypes.	97
4.3	The general design architecture of ONCOCIN.	101
4.4	A TNET showing alternating courses of chemotherapy.	102
4.5	Representation of concurrent events in TNET.	105
4.6	The structure of a TNODE object.	106
5.1	Architecture of model-based summarization.	117
5.2	A one-compartment drug-elimination model.	119
5.3	Prototypical versus patient-specific model predictions.	121
5.4	Multidose drug levels: prototypical versus patient-specific model predictions.	124
5.5	Interval abstractions derived from differences in model predictions.	128
5.6	The ONCOCIN record of a patient who has received MOPP chemotherapy.	135
5.7	TOPAZ-generated summary.	136
6.1	Graphical representation of the TOPAZ bone-marrow model.	140
6.2	State equations in the TOPAZ bone marrow model.	146
6.3	Predicted bone-marrow-model states using population parameter values. . .	150
6.4	Predicted WBC counts following multiple courses of MOPP chemotherapy.	151
6.5	Predicted white blood cell counts using patient parameters.	157
6.6	A seven-compartment bone-marrow model.	159
6.7	Changes in a parameter estimate with increasing observations.	161
7.1	The structure of a basic ETNODE.	168
7.2	An ETNODE with context-specific rules.	171
7.3	An ETNET RULENODE.	172
7.4	The ETNODE instantiation algorithm.	174
7.5	Minimum and maximum spanning intervals.	175
7.6	Temporal-searching diagrams.	176
7.7	Clinical abstractions represented in ETNET.	180
7.8	ETNODE intervals created by model prediction differences.	181
7.9	Causal explanations of surprising observations.	183
7.10	Critiquing the patient-specific model fit.	185
7.11	Temporal dependencies not detected by the temporal-searching algorithm. .	187
7.12	Abstraction of cyclic events in ETNET.	188
7.13	A two-table relational scheme to encode an ETNODE.	190
7.14	Patient-specific alterations to the expected clinical course.	192
7.15	Patient observations versus model predictions to define the true clinical course.	194
8.1	The TOPAZ rhetorical schema.	199
8.2	The PROSENET methodology for text generation.	201
8.3	A TOPAZ text-generation node.	203
9.1	Continuous and discrete temporal models.	215

CHAPTER 1

Introduction

Change haunts medical decisions. Pathologic processes disrupt vital body functions. Homeostatic processes oppose life-threatening instabilities. As time passes, pathologic, homeostatic, and therapeutic processes combine in perplexing ways. The patient's medical record catalogs the clinical signs and symptoms caused by these competing processes as they reveal their effects over time. Skilled physicians infer many critical clinical concepts, such as disease progression or therapeutic response, by interpreting changes observed during the course of a patient's illness. Temporal concepts are especially critical in medical therapy-planning. If given early enough, specific therapeutic choices can abort or suppress evolving undesired changes in a patient's clinical status. Effective medical decision making demands that we recognize and exploit the complex temporal changes that permeate the medical record.

This dissertation concentrates on representing and using medical knowledge about temporal relationships so that decision-support programs can deduce abstract clinical concepts that appear as temporal patterns in time-ordered clinical observations. A demonstration program, called TOPAZ,¹ illustrates the methodology proposed in this dissertation. TOPAZ generates a focused summary of the clinically important high-level concepts implied by the temporal patterns in a patient's medical record (Figure 1.1; Appendix C). A physician or a sophisticated computer-based decision-support system can use this summary

¹TOPAZ = Time Oriented Patient AnalyZer.

to modify decisions in light of temporal trends in a patient's clinical course. For example, a patient with a recurring problem who showed a poor response to the usual treatment on previous occasions is not likely to have a good response to the same treatment during the next episode. For medical therapy-planning problems, the options and choices for optimal, patient-specific treatment strongly hinge on the patient's *past clinical course*, *current clinical status*, and *predicted future course* (prognosis). The patient who did not respond to usual therapy on prior occasions is a candidate for more aggressive, but more dangerous, treatment, if his prognosis given continued therapeutic failure is poor. A therapy-planning system that creates the best possible patient-specific treatment plan must be capable of representing and reasoning about a patient's past, present, and potential future.

The work described in this dissertation focuses on temporal issues in medical decision making. Most real-world decisions involve reasoning in an incompletely understood changing context. Thus, there is a great need for intelligent systems to represent temporal relationships and to reason about the abstract temporal concepts implied by time-ordered observations. Because no special domain assumptions are made that could limit the applicability of this work to medical decision support, the issues and solutions presented in this research readily generalize outside of medicine.

This dissertation is based on three hypotheses that are central to the development of the methodology and the implementation presented in later chapters:

1. The diversity of temporal knowledge that is used in complex temporal problem solving demands multiple representational and inferential formalisms that work cooperatively
2. Explicit structural models that relate time-ordered observations to underlying concepts provide intuitive methods to interpret, explain, and predict subtle temporal patterns
3. Context-specific reasoning limits the size of the problem space by constraining the temporal features and relationships that must be considered in the current reasoning task

In this dissertation, I examine the role of temporal models and temporal reasoning in medical therapy-planning systems. I argue that previous computational models of time based on a single representational method are limited by the expressive power of the chosen method. I propose that combining multiple temporal formalisms facilitates the creation of

WBC X 1000	12.9	9.2	13.0	3.4	2.1	1.75	1.8	3.86	3.5
Chemo Name		MOPP	MOPP	MOPP	MOPP				
Mustard		11	11	11	8				
Vincristine		2.0	2.0	2.0	2.0				
Procarbazine		175	175	175	175				
Prednisone		80	80						
Day	10	16	23	12	19	01	09	15	14
Month	Apr	Apr	Apr	May	May	Jun	Jun	Jun	Jul
Year	87	87	87	87	87	87	87	87	87

This is a 21 y/o female who presented with clinical stage III-E-E-A, pathological stage I-A Hodgkins lymphoma. The patient's medical record begins on 10 April 1987 and ends on 14 July 1987. She was assigned to MOPP chemotherapy and received 2 treatment cycles.

Analysis of Clinical Course:

Based on our analysis of the entire clinical course, the best fit of our model to the data suggests the following key patient-specific physiological parameter values:

1. A steady-state bone marrow mass that is approximately 0.9 times the population value. This is considered a mild reduction in steady-state bone marrow mass.
2. A duration of drug effect that is approximately 4.2 times the population value. This is considered a moderate increase in the duration of drug effect.
3. A sensitivity to drug effect that is approximately 2.4 times the population value. This is considered a moderate increase in the sensitivity to drug effect.

Based on differences between population-based predictions (the expected clinical course) and the patient-specific model predictions (the modeled clinical course), the patient had one remarkable clinical event.

1. A series of 6 visits after starting chemotherapy, starting on 12 May 1987 and ending on 14 July 1987, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 15 June 1987 (35% of the expected value).

After adjusting for the patient's steady-state bone marrow mass, the two drug-related parameters, duration of drug effect and drug sensitivity, jointly caused an increased drug effect which led to increased myelosuppression and lower than expected WBC counts. The increased drug effect had a peak myelosuppressive effect 8.7 times greater than expected on 1 June 1987. The accumulation of drug due to the prolonged duration of drug effect was responsible for a 3.6 fold increase in drug effect. The increased sensitivity to drug was responsible for a 2.4 fold increase in drug effect.

Figure 1.1 Portions of a patient record and of a TOPAZ summary.

This is the hematologic and therapeutic portions of the medical record from a patient who has received MOPP chemotherapy. Also shown is a portion of the TOPAZ-generated summary for this patient. This summary and others appear in regular-sized type in Appendix C.

robust temporal models that solve a critical therapy-planning task: The abstraction and summarization of a patient's past medical record. This task requires a detailed understanding of *what the patient's data mean* rather than *what the patient's data are*. For an intelligent system to accomplish this goal, it requires a methodology that contains three features: (1) represents the medical knowledge about known temporal relationships, (2) detects the existence of these temporal relationships from a set of patient observations over time, and (3) summarizes raw time-ordered observations into salient abstractions that capture the clinical implications of the temporal trends. The TOPAZ program applies these principles to perform this summarization task.

A key element of TOPAZ is the simultaneous use of three representational techniques to capture different aspects of essential temporal knowledge and reasoning. Multiple representational methods permit each technique to represent those temporal relationships that are best expressed within a particular formalism. Combining more than one modeling method to enhance representational power is not a new idea, but the methodology proposed here combines numeric and symbolic models of time in a unique manner. As a demonstration program, the design goals for TOPAZ are

1. To illustrate the ability of combined models to capture temporal relationships that would be difficult to express in any single model formalism
2. To develop new techniques to combine disparate modeling methods into a cohesive, logical reasoning unit
3. To create a system, based on the proposed modeling methodology, that performs a difficult and useful task for medical therapy planning

Time is an integral part of any dynamic system. Because temporal reasoning and representation problems are central to this work, in Sections 1.1 to 1.4, I discuss the basic issues faced by designer of intelligent reasoning systems that must represent entities that cause change, and that must reason about the implications of those changes. In Section 1.5, I present an overview of the key research contributions of this dissertation to temporal representation and to temporal reasoning.

1.1 The Time Problem

Models are approximations of the real-world entities that they represent. A computer model of the real world is a set of symbols and operations. A computational model of time is constructed from the symbols and operations that represent temporal concepts within the computer. The choice of symbols, operators, and approximations limits the congruence of the model and the real-world entities. For one task, the mismatch may not be important; for another task, the mismatch may render the model useless. For example, a model of cardiac physiology (a functional model) may be adequate to solve problems of altered blood flow (a functional problem), but it probably will be unable to solve problems of altered anatomical relationships (a structural problem).

In this section, I focus on the concepts that must be expressed by the symbols and operations in a computational model of time irrespective of any particular formalism or implementation. I am concerned here not with the construction of such models, but rather with the concepts that should be captured by these models. Relevant work by other researchers is reviewed in Chapter 2.

This research is concerned with extending the ability of medical decision-support programs to reason about a patient's changing clinical course. This concern presumes that there are problems with reasoning about time in decision-support systems.² The time problem is not a unique property of the medical problem solving; it is a general problem that plagues all intelligent systems that reason about a changing world. For our purposes, the time problem is equivalent to the problem of reasoning about change. What is it about change that makes time a nemesis for intelligent systems?

A fundamental assumption in constructing intelligent systems is that the problem solver or reasoning component has an internal model or representation that captures the essential features of the problem world. The problem solver reasons by manipulating the elements of this internal model. An intelligent system cannot reason about features in the real world that are not represented within the internal model. The enterprise of constructing formalisms for expressing real-world entities in computer-usable internal models is called *knowledge representation*. The difficulties associated with time and change are problems in the available knowledge-representation techniques for constructing internal models that

²The use of the singular noun in the section heading is misleading but is in common use. Time presents many different problems. The reference to a singular time problem could imply there might be a singular time solution—which I sincerely doubt.

successfully and efficiently capture key properties of a dynamic, evolving real world.

The essential property of change is that *something* changes. Because change happens, something that was false, or did not exist, or was not relevant now becomes true, or exists, or is relevant. In artificial intelligence (AI), the problem of dealing with changing propositions has been called the *frame problem*. Initially described by McCarthy and Hayes in the narrow context of the situation calculus [78]—a particular formalism for reasoning about actions—the term has been generalized to stand for the problem of describing in a computationally reasonable manner what properties persist and what properties change as actions are performed and changes occur [48]. Hayes illustrated the frame problem using the following example:

When I go through a door, my position changes. But the color of my hair, and the positions of the cars in the streets, and the place my granny is sitting, don't change. In fact, most of the world carries on in just the same way that it did before, or would have done if I hadn't gone into room r1. But since many of these things CAN change, they are described in our vocabulary as being relative to the time-instant, so we cannot directly infer, as a matter of logic, that they are true in state *gothru(door,s)* just because they were in state *s*. This needs to be stated somehow in our axioms. In this ontology, whenever something MIGHT change from one state to another, we have to find some way of stating that it DOESN'T change whenever ANYTHING changes. And this seems silly, since almost all changes in fact change very little of the world. One feels that there should be some economical and principled way of succinctly saying what changes an action makes, without having to explicitly list all the things that don't change as well; yet there doesn't seem to be any other way to do it. *That* is the frame problem. [48, pg. 125]

A closely related problem with reasoning about change has been called the *qualification problem* [77]. All formalisms that represent change must express the same basic concept: *If this is true at time t_1 , then that is true at time t_2* [108]. The problem is specifying all the conditions in the *if* part to ensure that the *then* part is properly applied. Shoham illustrated the qualification problem in reasoning about a rolling ball:

To infer that a ball rolling in a certain initial direction will continue doing so, it is necessary to verify that there are no strong winds, that no one is about to pick up the ball, that the ball does not consist of condensed explosives that are about to go off in the next instant of time, and we can continue this list, getting arbitrarily ridiculous. [108, pg. 16]

Although it is possible to reason without stating these conditions explicitly, we may be making incorrect predictions if, in fact, one of the ignored conditions is true. The

tradeoff seems to be between the amount of knowledge that is required to make inferences and the accuracy of those inferences.

At first glance, we might assume that representations that encode the basic laws of physics, expressed as equations with time as an independent variable, escape the frame and qualification problems. Unfortunately, they do not. Most laws of physics embody simplifying assumptions, such as the absence of friction or the nondeformity of objects. Although simple laws can be extended to eliminate the most serious assumptions, at some level of modeling, simplifying assumptions creep into the formulae. The qualification problem requires that all potential complications be checked to ensure that the equations still hold. The list of potential invalidating conditions can get as extravagant as did that in the example by Shoham. Although we may have useful equations that describe the movement of two balls over a surface, we assume that there cannot be a third ball that suddenly appears between the two balls modeled by our equations. The frame problem requires us either to assume that our model is complete or to enumerate all possible complications that could occur during the movement of the balls. This list of complications would be enormous and could be ridiculous.

In Chapter 2, I shall present alternative formalisms for representing time and for reasoning about facts over time. In all approaches, the key issue is to capture the features that do and do not change as events occur. The simplest solutions invoke the frame assumption: *The only facts that change are those facts that I explicitly tell you have changed.* In this view, causality determines what is and is not affected and therefore what does and does not change. Depending on causality to solve the frame problem is not likely to be successful. Causal knowledge and temporal knowledge, although closely related, are not identical. Five key properties of temporal relationships that are independent of any implied causality are

1. Current assumptions are invalidated by later events
2. Current information is incorrect and is changed at a later date
3. Past information is incorrect and is changed at this time
4. Past events are never completely known
5. Past information can arrive out of temporal sequence

For an intelligent system to maintain an internal model that accurately reflects a changing state of knowledge about a changing external world, a solution to the time problem must address these uniquely temporal issues.

1.2 Dynamic Systems

Real-world dynamic systems are troublesome beasts. As does any system, they consist of a set of entities and a set of relationships that tie together the entities into a bona fide structure. The most troublesome feature of dynamic systems is the existence of *processes* that exert influence on entities over time. A proper definition of a process is elusive; much philosophical debate has not produced a universal definition.³ Despite our difficulties in defining what a process *is*, we can describe a dynamic system's change in behavior because of what a process *does*. Our goal is to model accurately, with symbols and operators, the effects of processes on the entities within a system. When multiple processes coexist and interact on the same entity, separating out the effects of each process is more challenging. The problem becomes especially difficult to solve if process effects do not follow the superposition principle. The *superposition principle* states that the effect of two processes is the sum of the effects of each process alone. For example, suppose a system has two inputs, A and B. With input A, the output is A' ; with input B, the output is B' . If the system obeys the superposition principle, the system output with both inputs A and B present is $A' + B'$. Without the superposition principle, the final output with both inputs is some function more complex than the simple sum of the individual outputs. The loss of superposition means that the two inputs are intertwined, producing an effect that cannot be predicted from knowledge of each process independently. Other complex, nonlinear interactions—such as triggers, thresholds, and saturation effects—further complicate the range of behaviors that real-world dynamic systems can exhibit. In medical physiology, these nonlinear behaviors are common.

Mathematics has a long tradition of constructing formal models to represent dynamic systems. Advanced numerical techniques have been developed to model complex

³That a formal definition of a process has been the subject of much philosophical debate is not a good sign. Philosophers disagree on the essential features that define a process and the attributes that describe a process [101,95,103,125]. There even is disagreement whether a process exists as a first-class entity or only as a relationship between entities. Fortunately, resolution of these deep issues is not required to meet our limited goals.

nonlinear interactions. When sufficient detailed knowledge exists about the relationships among entities in a real-world system, numeric models can closely predict observed temporal behaviors. Entities of interest within the dynamic system are expressed as variables; relationships among entities are functions of variables. Although specialized mathematics have been developed to capture complex behaviors, all numeric mathematical models consist of variables and functions. Time usually is an explicit independent variable that is used in functions to describe changes in system characteristics. Numeric models are often used (1) to predict the value of a variable at some point in time, (2) to estimate the value of a model variable when given a set of system observations, and (3) to adjust a set of control variables to achieve desired values on a set of goal variables. Prediction, estimation, and control are tasks for which the most sophisticated and successful mathematical techniques exist.

Given that numeric mathematical techniques are able to capture the temporal behaviors of complex dynamic systems, what are the outstanding issues in temporal reasoning and representation? In Section 1.1, I discussed the implicit presence of the frame problem in the simplifying assumptions that underlie numeric formulae that model ideal situations. For knowledge acquisition, the precision and detail that must be elicited from the domain expert, if the numeric methods are used, pose a significant problem. Numeric models require minute details which may be difficult to obtain. An area of active research, called *qualitative modeling*, is attempting to develop new modeling techniques that eliminate the need for the model builder to commit explicitly to every detail, allowing him to specify only those system details that are known [16,41,42,63,62,64,131]. In this dissertation, I am concerned with a related but different problem. Effective and efficient reasoning frequently occurs at a higher level of abstraction than is represented in complex numeric models. Traditional numeric techniques offer no insight into how to map from the low-level details required by and returned from numeric methods to the abstractions and language used by people.

1.3 Temporal Inferences from Time-Ordered Data

Data are quantitative and qualitative statements about something of interest in the real world. For this work, the *something of interest* is a dynamic system. Data may describe a feature of a single entity, called an *entity attribute*, or a feature of a collection of entities, called a *relation attribute*. The set of all entity- and relation-attributes defines

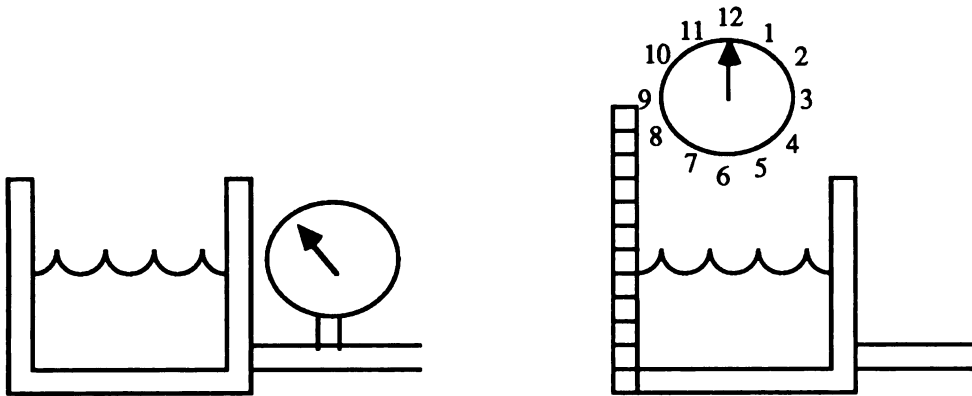


Figure 1.2 Two methods for determining a flow rate.

The system on the left has an instantaneous flow-rate meter. The system on the right has no flow-rate meter but has a level gauge and a clock. In the system on the left, we measure the flow-rate value directly from the meter; in the system on the right, we *infer* the flow-rate value by measuring the change in fluid level over a timed interval. In the second system, the passage of time is essential to infer the unmeasured system attribute (flow rate).

a *system state*; the set of all entity and relation attribute values defines a system's *state vector*.

In real-world systems, the set of available data never specifies the complete state vector. Many essential system attributes are not measured, either because they are not readily accessible or because no measuring device exists to describe a given attribute's value. For example, although an instantaneous flow or rate constant could theoretically be measured by a flow meter, practical considerations may result in inferring the rate-constant value indirectly by dividing the difference in two level values by the time elapsed between the two measurements (Figure 1.2). In this example, the passage of time is essential if we are to infer the value of the unmeasured rate constant by observing its indirect effect on a measured attribute (changes in a timed series of level measurements).

Time-ordered observations taken from a dynamic system contain meaningful temporal patterns among observations, such as the change in a level due to an inflow or outflow. Many unobserved system attributes and relationships exert their influence only with the passage of time. These hidden relationships cause distinct temporal patterns that a knowledge-

able interpreter can use to make sophisticated inferences about unobservable, temporally sensitive attributes. Comprehensive knowledge about the temporal relationships within the observed system are essential to understanding the constraints and patterns exhibited in the time-ordered data.

1.4 Context-Sensitive Reasoning

The number of potential relationships among a large time-ordered data set is enormous (Section 2.1.4). Only some temporal patterns are realized in a physical system, and even fewer patterns have meaningful implications. Although a dynamic system may have a large number of potential attribute values and behaviors, at any particular instant, only a very small number of them are immediately attainable. For example, although an integer-valued variable may potentially vary between 0 and 100, if the current value is 50 and it is known that the variable cannot change by more than ± 10 percent over a short time interval, then the range of possible values for the variable in the immediate future is substantially restricted.

The current system state limits the range of next states that are readily accessible. I shall use the word *context* to denote a set of system states that constrains the system to a small set of possible behaviors. The features that relate a collection of system states to an identifiable context are not precisely specified because, as I shall discuss in Chapter 5, the nature of the problem-solving task influences which features constitute a useful context. Context provides a powerful mechanism to constrain the amount of reasoning that needs to be applied at a given moment. Without contextual constraints, total knowledge about all possible system states must be present. With the proper contexts, only those relationships that have immediate bearing on the current behavior of the system need to be examined. Context-specific reasoning can be an essential focusing mechanism that provides significant reasoning efficiency by decomposing time into pertinent aggregates.

The use of context to constrain reasoning is not without concomitant problems. Assuming an incorrect context may result in using an incorrect set of constraints and methods. Inferring the correct context may be difficult and may require information that is not found in the available data. For example, an electrical system may show periodic deviations. Without the additional information that excessive power drains occur when the elevator is used and that the electrical system is sensitive to this power drain, it would be impossible

to know that this deviation constitutes a reasonable context that has its own assumptions that differ from contexts in which the power drain is not present.

If a context defines a set of assumptions, constraints, and methods that apply when the context is in effect, then the presence of multiple contexts yields multiple sets of these features. Multiple simultaneous contexts present difficult reasoning issues because they can potentially act in synergistic or antagonistic ways. For example, the presence of one context can invalidate certain features of a second context. Some constraints may be relevant only when both contexts are present. In general, any set of contexts can impose features that are not present when any one of the constituent contexts is absent. The presence of interactions among contexts implies that a context does not supply an invariant set of assumptions, constraints, and methods; rather, a context supplies a set of features that are conditional on the presence or absence of concurrent, interacting contexts.

The term *context-specific reasoning* refers to the process of changing assumptions, constraints, and reasoning methods in response to changes in system contexts. The purpose is to make reasoning more effective, such that only those assumptions and reasoning methods that apply to the current context are used. The methodology proposed in this dissertation heavily exploits the concept of context-specific reasoning.

1.5 Overview of Key Contributions

In this section, I shall present the major research results of this dissertation. The results are presented here only in skeletal form. My purpose is to sketch the general landscape of the research. It is my intention that this early overview of the key contributions give the reader a general perspective of the detailed work that is presented in later chapters.

This work is motivated by recurring problems in temporal reasoning and representation for medical therapy-planning systems (Chapter 2). In this problem domain, temporal concepts play an important role in determining the proper treatment plan. I use computer-based patient medical records, in flowsheet format, as a source of time-ordered clinical observations (Figure 1.3; Appendix C). This simple time-ordered input contains significant implicit temporal features that an intelligent agent cannot detect by examining single observations. Because temporal features can alter therapy decisions, these concepts need to be represented and inferred for use in medical decision making. The goal is to formulate a methodology that supports the expression of temporal constraints, the detection

of temporal features, and the representation of temporal findings for use in knowledge-based systems. The proposed methodology is demonstrated in a prototype system that analyzes a patient's medical record for significant temporal findings, and that constructs a summary of the patient's evolving clinical course (Figure 1.1).

1.5.1 Combining Quantitative and Qualitative Temporal Models

Physiological implications are the essential insights that must be derived from primary clinical data for medical therapy planning. I use a numeric model to represent the relationships among physiological entities (Figure 1.4). This numeric model, consisting of a set of continuous differential equations, captures the *processlike* features of time. Mathematical fitting techniques are used to estimate patient-specific values for model parameters from patient observations. This is a fairly standard use of physiological models and parameter estimation. Unfortunately, these techniques take a series of time-ordered numbers (clinical observations) and return another series of numbers (atemporal model-parameter values and time-ordered model-state predictions). The new set of numbers represents underlying physiological features, rather than superficial patient observations. The task remains to summarize the temporal patterns in the physiological estimates into clinical language that captures the implications of the physiological changes in an intuitive and natural style.

A second temporal model, using symbolic methods, constructs abstractions of the patient-specific model parameters and time-ordered model predictions by detecting intervals of time during which interesting findings or predictions were present (Figure 1.4; Section 1.5.2). This model captures the *eventlike* features of time. Abnormal parameter estimates trigger the creation of symbolic nodes that represent the clinical implications of a parameter's numeric value. The symbolic model generates higher-level abstractions by creating additional nodes that aggregate lower-level clinical abstractions represented by simpler nodes (Section 1.5.3).

A third temporal model, using symbolic methods, builds the final summary text from the abstraction intervals generated by the previous model (Figure 1.4). For each remarkable abstraction present in the data, the text-generation model formulates statements that describe the abstraction in the proper context in which it occurred. This model captures the *statelike* features of time. Comments are organized around specific patient states that were detected by the abstraction model. In this way, a focused, coherent description of the

Upper Nodes	6		4	3					
Lower Nodes	2		2	2					
Spleen									
Splenectomy									
Liver Bx									
Marrow Bx	1								
LAG	1			1					
IVP									
CXR	6		4	3					
Bone Scan	2								
CT Scan	6								
OVERALL	6			3					
Karnofsky	100		100	100					
Weight	64.8	64.8	64	66.9	64.4				
WBC X 1000	12.9	9.2	13.0	3.4	2.1	1.75	1.8	3.86	3.5
PCV	32.1	30.5	36.3	32.9	32.5	29.0	31.2	34.9	34.5
Hemoglobin	10.7	10.1	12.1	11.2	11.1	10.6	10.8	11.8	11.7
Platelets	563	498	473	207	201	245	262	218	164
BSA (m ²)		1.81							
Chemo Name		MOPP	MOPP	MOPP	MOPP				
Cycle #		1	1	2	2				
Subcycle		A	B	A	B				
Visit Type	LAB	TX	TX	TX	TX	LAB	LAB	LAB	LAB
Mustard		11	11	11	8				
Vincristine		2.0	2.0	2.0	2.0				
Procarbazine		175	175	175	175				
Prednisone		80	80						
Port									
RTx Start Date									
RTx End Date									
Cum. # Fraction									
Cum. Max. Dose									
Number Rads									
RT Toxicity									
Day	10	16	23	12	19	01	09	15	14
Month	Apr	Apr	Apr	May	May	Jun	Jun	Jun	Jul
Year	87	87	87	87	87	87	87	87	87

Figure 1.3 A sample patient flowsheet.

This is one page of the flowsheet record for the patient in Figure 1.1. The complete flowsheet includes a second page of measured attributes and a third page of identifying patient information (name, address, diagnosis). New information is added to a blank column on the right end (not shown).

notable events can be constructed.

The use of multiple modeling methodologies is central to the detection and abstraction of temporal implications in the medical record. Each method is used to represent concepts that are best expressed within that formalism. Numeric methods express physiological concepts well but are not capable of generating output that resembles clinical language. Symbolic models capture changes in numeric values and convert them into clinical language. Symbolic models support abstraction and summarization techniques that would be difficult to emulate within a strictly mathematical model. When we combine these two methods into a well-integrated system, we can accomplish tasks that are not possible using either method alone.

1.5.2 Temporal Extensions to the Time-Oriented-Data Model

The time-oriented-data (TOD) model is a simple tabular database model where tuples are $\langle \textit{attribute}, \textit{date}, \textit{value} \rangle$ triples [129,44].⁴ The value of an attribute on a date or range of dates is readily available. I have extended this model by adding an object-oriented superstructure, called TNET, that represents intervals of contiguous time (Figure 1.5). An object denotes an interval of time in which some event or context was in effect. The definition of an *event* or *context* is not restricted; any concept that has meaning to the knowledge engineer within the problem domain can be represented. A TNET object has two states—open or closed. An *open* TNET object represents a context that is still in effect; a *closed* TNET object represents a context that has concluded. Hierarchies of objects express the concepts of temporal dependency and concurrency. I model temporal dependency—defined as the constraint that an event lower in the TNET hierarchy cannot remain open if the superior temporal object has been closed—by closing all dependent children TNET objects when the parent TNET object is closed. I model concurrency—defined as the presence of more than one nondependent context—by multiple hierarchies projecting onto the same time line. Because of concurrency, TNET objects may be shared by more than one hierarchy. A shared TNET object expresses the concept that the event represented by that object occurred while concurrent events were happening.

TNET supports a temporal query language, called TQUERY, that implements date-free, context-specific queries. Complex temporal queries that either could not be for-

⁴The original formulation of the TOD model has $\langle \textit{patient}, \textit{attribute}, \textit{date}, \textit{value} \rangle$ tuples. The patient key can be dropped when only a single patient record is considered.

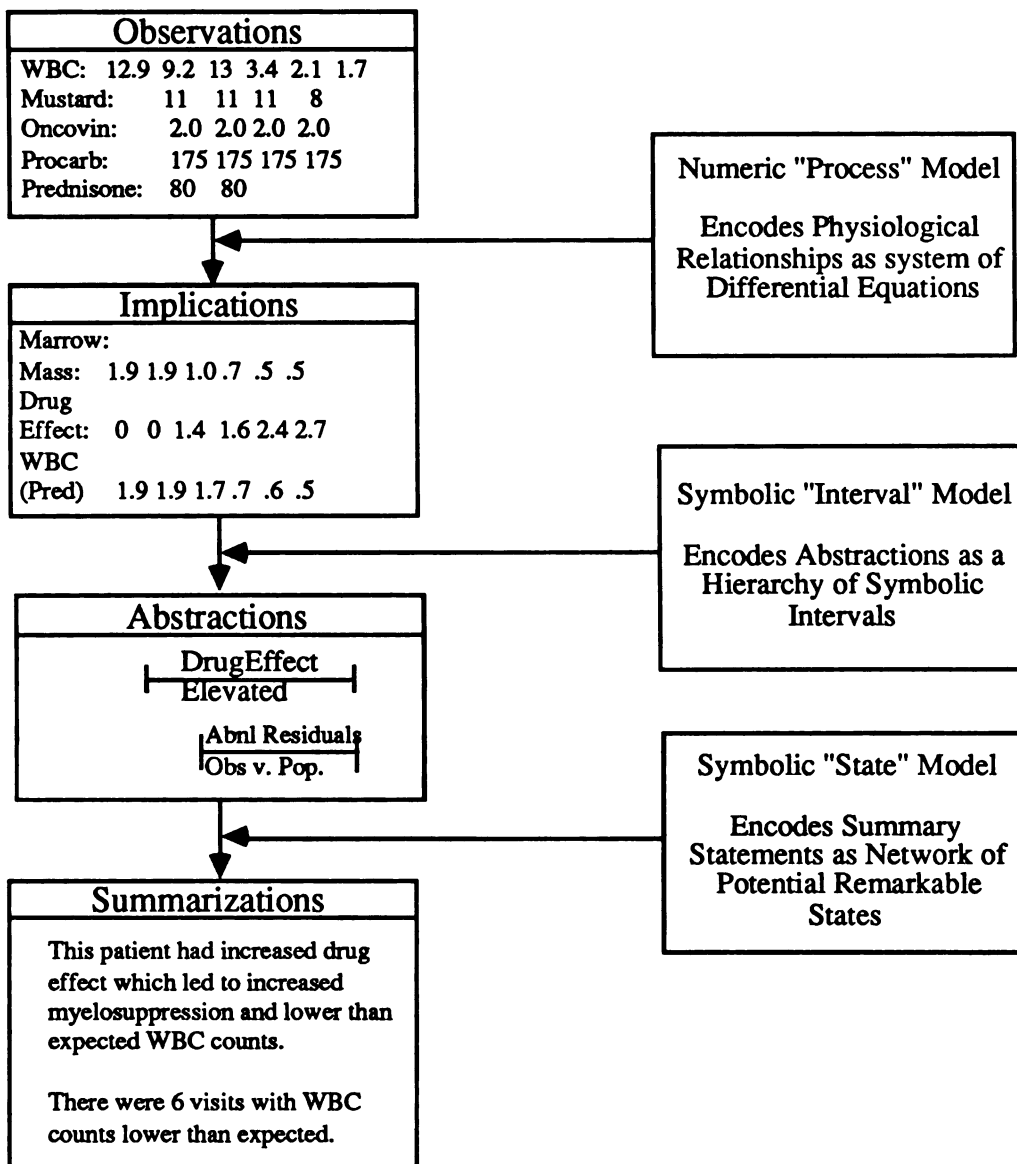


Figure 1.4 The use of multiple temporal models for temporal data summarization.

TOPAZ is a program designed to summarize time-ordered clinical data. It embodies three models of time: (1) a process-oriented temporal model encoded as a set of numeric differential equations, (2) an interval-oriented temporal model encoded as a set of symbolic intervals, and (3) a state-oriented temporal model encoded as a symbolic transition network. Each model encodes temporal features that would be difficult to capture using the other formalisms.

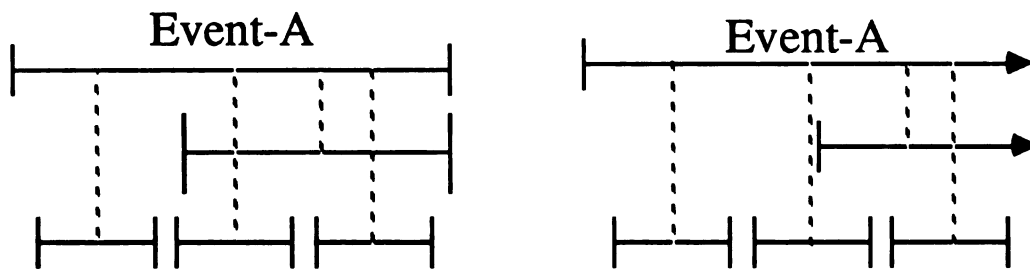


Figure 1.5 A graphical representation of a temporal network.

This is a temporal network (TNET) that demonstrates contextual dependency. Time passes from left to right. Intervals with arrows on the right are “open”; all other intervals are “closed.” A parent–child relationship, depicted as a dashed line between temporal intervals, denotes a contextual dependency between the two intervals. More complex TNET structures are presented in Chapter 4.

mulated or were computationally expensive due to linear searching are processed efficiently via traversal of the temporal-network superstructure to obtain context-specific tuples.

TNET is a static structure that supports database queries. ETNET, an extended reimplementaion of TNET, has three new features to combine context-specific reasoning and context-specific querying capabilities within a unified structure (see Section 1.5.3):

1. ETNET permits reasoning methods to be associated with each interval object. This extension is used to implement context-specific reasoning. These reasoning methods are available to an inferencing system only while the interval is open. This technique implements context-specific reasoning where new intervals add new reasoning methods that make sense only when the event represented by the interval is in progress.
2. ETNET permits future intervals—interval events that start and stop sometime after the current date. A distinguished future date permits the representation of intervals that continue either indefinitely or until some future condition is satisfied. In TNET, open intervals remain open until they are explicitly closed, but intervals that both start and stop in the future cannot be expressed.
3. ETNET supports possible-worlds structures, in which alternative sets of contexts symbolize competing representations of the past, present, and future.

ETNET capabilities are a superset of TNET's that allows representation of more complex temporal relationships. The incorporation of reasoning methods in the temporal network expands the role of the network from a static structure that supports difficult temporal queries to an active element that participates in context-specific temporal reasoning.

1.5.3 Context-Sensitive Reasoning

ETNET objects represent contiguous intervals of time in which a particular context was present. The term *context* is used loosely and is intended to mean *an interval of time during which something of interest was happening*. Events such as the delivery of chemotherapy, the simultaneous delivery of radiation therapy, or the delay of treatment are obvious interesting temporal intervals that are represented by separate ETNET objects. For the interpretation task, a steady drop, a sudden change, or a plateau in a sequence of parameter values may be interesting and may trigger the instantiation of an ETNET object. Each object type contains a set of instantiation rules that defines the conditions that must exist for an instance of the object to be created. When an object type is instantiated, it places new rules onto an active rule queue. The added rules can specify methods to conclude attributes associated with the new ETNET object, methods to instantiate other ETNET objects, and methods to terminate the ETNET object. When an ETNET object terminates, the rules that were added to the rule queue by the interval when it was created are removed. An instantiated interval represents a context that is still current, whereas a terminated interval represents a context that is no longer valid. As contexts are instantiated and terminated, the set of rules and the conclusions that can be drawn by these rules constantly change over time. Only those rules that make sense given the current set of instantiated intervals are found in the rule queue. When a context no longer applies, the interval is terminated, so that inappropriate context-specific rules are no longer available for execution.

ETNET generates clinical abstractions by having low-level intervals detect simple trends in single parameter values. These low-level intervals contain instantiation rules that, if triggered, create more global intervals, which represent more complex trends. Initial abstractions look for patterns among single parameters. In turn, the single-parameter abstraction intervals contain instantiation rules for even more global intervals that detect trends across multiple parameter intervals. The highest-level intervals represent the most

abstract concepts that can be concluded by the system. Thus, a series of time-ordered observations is converted into a network of interval-based objects that represents multiple levels of temporal abstractions of the data.

1.5.4 Summary

In this chapter, I have presented the three research hypotheses that shape the issues and methods to address the temporal reasoning and representation problems discussed in this work. Three main themes will reappear in later chapters: (1) multiple temporal models that work cooperatively must be used to solve temporal problems that cannot be solved by any single model, (2) an explicit model of the underlying system structure can be used to reason about the implications of superficial observations, and (3) context-specific reasoning must be used to limit the amount of available knowledge to only as much as makes sense in the current problem-solving context. Time presents difficulties for designers, both of database systems and of knowledge-base systems. This work gives new methods and tools to both communities.

CHAPTER 2

Temporal Models in Medical Decision Support

The clinical environment is a complex setting for decision making. Time is only one feature that complicates medical decision making, but our innate ability to manipulate subtle temporal concepts engenders a false sense of complacency toward building models of clinical temporal reasoning. The concepts and relationships that can be used by a computer-based reasoning system strongly depend on the ability of a model encoded within the computer memory to capture the essential elements of the clinical problem. To derive meaningful conclusions in a changing medical setting, medical decision-support systems must represent the temporal nature of the clinical environments they attempt to model.

There are numerous competing modeling approaches that attempt to express temporal knowledge and dynamic change. Each approach proposes a set of primitive concepts and relationships to construct models of the real world. Computational models are usually described in terms of model expressiveness and computational tractability. *Model expressiveness* refers to the breadth of concepts that can be represented using the model formalism. *Computational tractability* refers to the ease with which the reasoning system can derive implications from internal model structures. A modeling methodology that is expressively rich but computationally intractable is not useful in constructing real decision-support systems.

Model expressiveness and computational tractability tend to be opposing characteristics of a modeling methodology. Computational models of time must make tradeoffs in semantic and representational power. In this chapter, I look at the semantic and representational differences between two major classes of temporal models: (1) modeling approaches that emphasize time as the primitive concept, and (2) modeling methods that represent process and change as the primitive concepts. In the first approach, the notions of process and change are constructed from temporal primitives. In the second approach, time is implicit in the definitions of the process primitives. The ontologic differences between the two paradigms vividly demonstrate the tradeoff in expressiveness made by the adoption of a particular approach to representing time. One of the three hypotheses central to this work is that the complexity of temporal reasoning and representation demands the use of multiple modeling methodologies. By examining the advantages and disadvantages of both the time-based and the process-based models, I will show that there are concepts that can be represented in one formalism but not in the other. This observation is important to the implementation of the TOPAZ program which, as proposed in hypothesis 1, uses multiple temporal models to incorporate key features from both classes of temporal models.

Researchers in specific fields within computer science have recognized the special role of time. Each field has proposed solutions to various temporal issues using terminology and techniques that are familiar to the specific research community. This fragmentation of related work has lead to duplication of effort, inconsistency in terminology, and limitations in transferring insights from one area to another. Fortunately, these unnecessary barriers are breaking down, a trend that is reflected in the interdisciplinary nature of this dissertation and in the increasing interest in the hybrid field of intelligent databases. Because of this prior fragmentation, I examine three approaches to computational models that are directly relevant to TOPAZ and to medical decision making: (1) computational models of time and change, (2) computational models of temporal data, and (3) computational models of decision making. This separation is admittedly artificial; most decision-support system have features that cross these boundaries. It is particularly useful, however, because the problems and solutions have differed significantly for each area.

The research topics reviewed in this chapter were selected for their immediate relevance to the development of the temporal data-interpretation methodology and to the implementation of TOPAZ. Time-related research from other fields is not examined. Important theoretical research topics that are not represented include philosophical arguments

on the structure of time [101,103,95,125] and linguistic analyses of the use of temporal references in spoken and written language [54,96,128]. These fields contribute additional insights and perspectives to the role of time in complex settings. Work in these fundamental areas may provide new models of how people conceptualize temporal abstractions. For example, a careful examination of the use of temporal references in natural language could provide new temporal ontologies that result in different computational approaches.¹ In addition, projects that attempt to construct natural-language interfaces with temporal features to databases have been excluded [20,76,75].

2.1 Computational Models of Time

People manipulate complex temporal facts without much apparent difficulty. We all have a set of notions about the general characteristics of time that enables us to remember the past, deal with the present, and plan for the future. The ease to which people deal with complex issues such as time in daily discourse suggests that there are conceptions about the world that are shared by most people. These intuitive concepts, which we use effortlessly in daily living, have been called “commonsense knowledge” by researchers who have been attempting to understand the nature of this knowledge [51]. Computational models of time attempt to capture these commonsense temporal notions in a format that can be represented in computer memory and manipulated by computer programs.

Computers are physical symbol processors. The symbols in a computer memory may represent numbers, letters, or concepts. Newell [94] proposed the notions of a *knowledge level* and a *symbol level* to distinguish between the knowledge used to exhibit intelligent behavior and the symbols used to encode that knowledge. A frequently unstated assumption is that symbols from the symbol level are an adequate medium for expressing knowledge from the knowledge level. Brian Smith clearly articulated this assumption in his *knowledge representation hypothesis*:

Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such external semantical attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge. [111, pg. 33]

¹My thanks to Bonnie Webber, who brought this possibility to my attention.

A domain-independent computational model of time does not exist. Philosophers have searched for a universal logic of time for centuries. The modern study of temporal and tense logics began with the classic book by Prior [101] (See also [103,95,125]). There are two opposing perspectives about the nature of time that philosophers, logicians, and computer scientists have used to justify particular models of time. One perspective assumes that time is a real entity with objects and properties that are unique to it. This perspective leads to models where time itself is the essential element to be characterized and manipulated. The opposing perspective argues that time exists only because of the existence of change; therefore, a model of change is in fact a model of time. This second approach yields models that express temporal concepts in terms of processes and the changes these processes cause. Time is an artifact or side effect rather than a primary object. The distinction is important for this work because the choice of perspective has a major influence on the terms, relations, and operations that are provided within a model language. For this reason, I briefly examine representative work from both viewpoints. TOPAZ maintains two models of the clinical environment; a physiological model (a process-oriented model) and a clinical-event model (an time-oriented model). By incorporating two models, TOPAZ represents temporal features found in both time-based and process-based temporal models.

2.1.1 An Interval-Based Model of Time

In this section, I examine a representative formalism that models time as the primitive concept. Formalisms of this type differ in the unit of time that is considered the basic temporal entity. The two main distinctions are models that consider instantaneous points or moments as the temporal primitive versus models that consider intervals or durations as the temporal primitive. The choice of temporal points or intervals is an ontological question. Recently, researchers have been exploring the differences in expressibility and computability between the two types of models [127,121]. Logicians have studied point-based systems more extensively than they have studied interval-based systems. Point-based temporal models have been proposed and implemented [79,32,33,34]. Interval-based implementations are more common in planning and natural-language-understanding programs. TOPAZ uses an interval-based representation to construct a model of the key events that occur during a patient's clinical course. Therefore, I present an example of an interval-representation formalism.

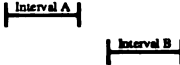
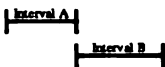
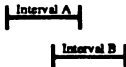
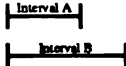
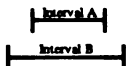
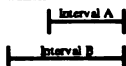
Allen defined a temporal calculus based on an interval representation of time [3, 4,7]. This calculus was subsequently reformulated as an axiomatic first-order theory by Allen and Hayes [5,6,49]. Properties and extensions to the Allen and Hayes logic have been explored by Ladkin [68,67,69], Vilain [126,127] and Tsang [121]. The logical system is simple, consisting of only temporal intervals as primitives and a single relation MEETS. MEETS is defined as the situation where there is no time between two intervals, and no time that the intervals share. There are 13 possible relationships that can exist between two temporal intervals (Table 2.1). Temporal ambiguity or uncertainty is denoted by more than one relationship being assigned to a pair of intervals. For example, if we know that interval A began sometime before interval B, but we do not know when interval A ended, we have incomplete knowledge about the precise temporal relationship between the two intervals (Figure 2.1). Using the symbol denotation shown in Table 2.1, we depict this temporal uncertainty by listing all temporal relationships that satisfy our partial knowledge. In this example, the notation, $A - (< m o f i c) - B$, encodes the fact that, with our current knowledge, interval A either BEGINS, MEETS, OVERLAPS, FINISHES BY, or CONTAINS interval B. If we later obtain new information that interval A ended no later than when interval B started, we can use this additional knowledge to remove relations that no longer satisfy this new information. In this example, we specify our new knowledge as $A - (b m) - B$.

With this interval notation, Allen defined a 13 by 13 matrix that gives the transitivity rules among triplets of intervals. For example, if $A - (<) - B$ and $B - (oi) - C$, the transitivity table specifies that $A - (< o m d s) - C$. These transitivity rules provide a method of calculating the transitive closure over a set of related intervals as new facts and intervals are added to a database of intervals. By determining the transitive closure over all interval triplets, we can infer new temporal relationships as new information becomes available. Using well-known techniques of constraint propagation and the transitivity table, we can easily implement Allen's interval calculus [126,53,52].

2.1.2 A Mechanism-Based Model of Processes

Qualitative reasoning about physical systems is a large and active area of research in artificial intelligence. Researchers are attempting to define modeling formalisms that provide qualitative predictions of a physical system's behavior based on qualitative descrip-

Table 2.1 Thirteen possible relations between two temporal intervals.

Relation (Symbol)	Example
A equals (=) B	
A before (<) B B after (>) A	
A meets (m) B B met by (mi) A	
A overlaps (o) B B overlapped by (oi) A	
A starts (s) B B started by (si) A	
A during (d) B B contains (c) A	
A finishes (f) B B finished by (fi) A	

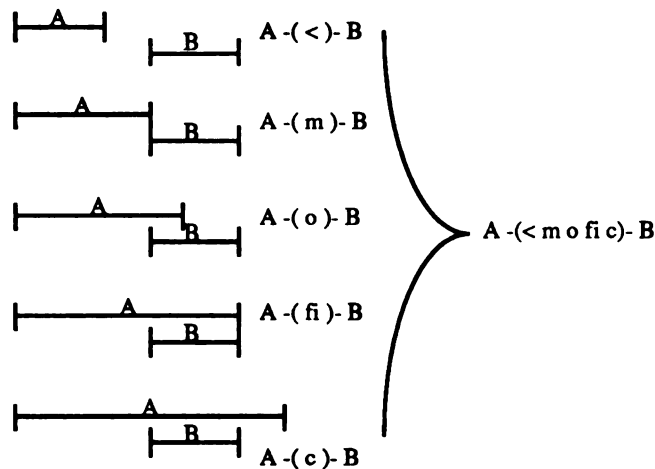


Figure 2.1 Representation of temporal uncertainty in the interval calculus.

We are told that interval A begins before interval B, but are given no information about when interval A ends. Shown in the figure are the five interval relations that satisfy our current partial knowledge. When we receive additional information, interval relations that no longer satisfy our new state of knowledge are removed.

tions of the system's structure [16]. Kuipers has focused on modeling mechanisms with continuous, time-varying parameters [63,61,64]. Other methods are being pursued by other investigators, but Kuipers' methodology of *qualitative differential equations* is especially relevant to the set of quantitative differential equations that constitutes the physiological model in TOPAZ.

In a Kuipers' qualitative model, a system is a set of variables and a set of constraints among pairs or triplets of variables. A variable has two descriptors: (1) a set of *ordinal relations* (such as $>$, $<$, and $=$) that hold between pairs of related variables; and (2) an *IQ value*² representing the sign of the derivative. The temporal behavior of a system is specified in a collection of constraints among time-varying, real-valued parameters. Three constraint types describe the qualitative relationships between pairs of system parameters: [64]

1. *Arithmetic*: ($X = Y + Z$). The value of the parameters must have the indicated relationship at each point in time.
2. *Functional*: ($Y = M^+(X)$). Y is a strictly increasing (monotonic) function of X (decreasing if M^-). M_z^+ and M_z^- pass through the origin as well.
3. *Derivative*: ($Y = dX/dt$). At each point, Y is the rate of change of X .

A simple physical system represented as a qualitative model is shown in Figure 2.2. A model in Kuipers' representation is a weaker level of description than is a numerical solution, results in less constrained system predictions than does an analytical solution, but still captures the process relationships among entities in the explicit constraints among system variables.

Kuipers defines a *time-point* as a qualitatively distinct state of the system. If there are no state changes (for example, there are none in a system at equilibrium), then there are no time-points created in these models. A time-point is associated with the set of model variable assignments that identify the distinct system state. Because of the definition of a time-point, each time-point may correspond to either an instantaneous moment or an interval of time during which the system state remained qualitatively the same. Kuipers notes that the number of time-points generated by the model is not due to the passage of time, but is related to the amount of detailed information that is available about the

²IQ value = incremental qualitative value.

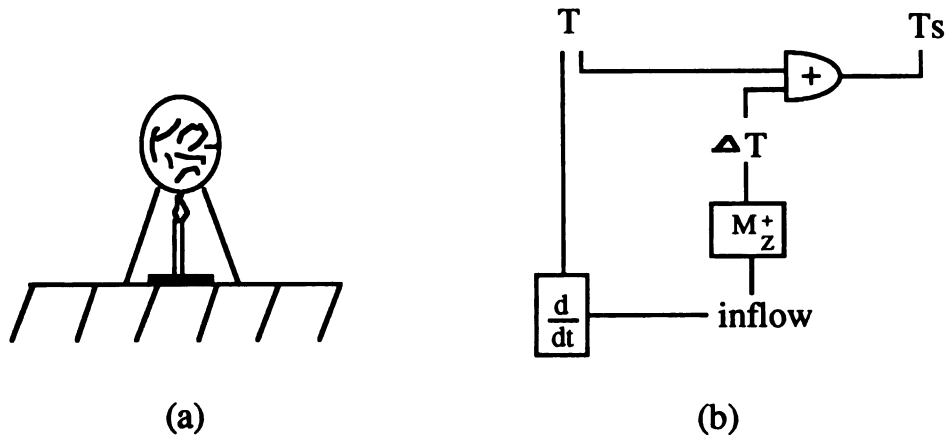


Figure 2.2 A physical system represented as a qualitative model.

(a) A schematic drawing of a container of gas is subjected to a constant temperature heat source. (b) The qualitative network representation of this physical system. (Adapted from [61, pgs. 175, 177])

system. As a more elaborate system model is constructed, more system-state changes will be distinguishable and detected by the model. It follows that more time-points will be generated by a more detailed model of the mechanism.

2.1.3 Ontologic Differences

I have presented only the essential features of Allen's interval-based model of time and of Kuipers' process-based model of mechanisms. Both methodologies are designed to model a world that changes over time. The key point is the striking differences in primitives, relations, and operators contained in each modeling approach, which reflect a markedly different ontology of time.

Allen proposed his calculus of temporal intervals to represent the duration of a property or fact about the real world. For example, Figure 2.3 depicts the changes in true facts as a person moves from one room to another in a house. In this example, a person leaves the living room, turns on a light as he enters a dark bedroom, and turns off the light as he leaves the room. Based on these facts, the interval containing the proposition that the person is in the living room (interval LR) MEETS an interval containing the proposition that he is in the bedroom (interval BR). Because he turns on the light when he enters

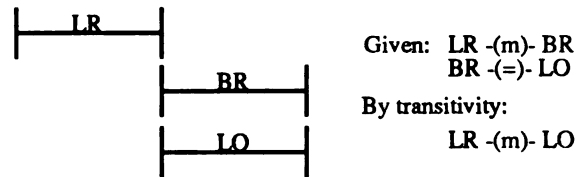


Figure 2.3 Interval representation of changing facts.

Interval LR represents the proposition that a person is in the living room. Interval BR represents the proposition that he is in the bedroom. Interval LO represents the proposition that the bedroom light is on. Propositions are true only over the duration of an interval that represents the proposition. In this figure, we are told the person leaves the living room and enters the bedroom. He turns on the bedroom light when he enters the bedroom and turns off the bedroom light when he leaves. We can infer that he turns on the bedroom light when he leaves the living room using the transitivity table.

the bedroom and turns off the light when he leaves the bedroom, the interval containing the proposition that the bedroom light is on (interval LO) EQUALS interval BR. Using Allen's transitivity table, we can infer that interval LR MEETS interval LO although this was not stated explicitly. Mechanisms or processes, such as the flow of electricity into the bedroom light bulb, cannot be directly represented using Allen's formalism. These concepts are captured indirectly only if they cause a fact about the world to change. In Allen's formalism, change is represented by a sequence of intervals generated by the succession of new facts about the world that are asserted by the effects of a process.

Kuipers' formalism indirectly represents time as a sequence of distinguishable system states that define *time-points*. It is not possible to represent or reason about the time between system-state changes. Like Allen's interval representation, assertions about the world in a Kuipers model are true until a state change occurs. For a new state to exist, a state change must add new facts and remove old facts. Unlike in Allen's models, the user has no direct control over the generation of *time-points*. They are created by the dynamic behavior of the model components. A second major distinction is that Kuipers cannot represent the same moment in time as two distinct *time-points* whereas Allen permits two intervals, containing different propositions, to entail the same temporal interval (for example, interval BR and LO in Figure 2.3). Kuipers directly models the temporal relationships

among components in a dynamic system as a series of constraints. This approach is identical to the classical quantitative formulae that Kuipers wishes to simulate qualitatively. In both classical quantitative models and Kuipers' qualitative models, time is an independent variable rather than a represented object. Because time is a quantity that is explicitly independent of all other quantities, there is no way to represent the temporal relationships other than by the effect of process relationships among related entities.

The ontologic differences between Allen's model of time and Kuipers' model of processes reflects different modeling goals for each methodology. Allen attempts to capture the temporal relationships between facts in the world; Kuipers wishes to capture the temporal effects of processes on objects in a mechanism. In medical problem solving, both perspectives make sense. The patient's past medical history consists of facts that changed over time as the patient's illness evolved and responded to previous therapies. This perspective fits well with Allen's view of temporal representation. Conversely, disease manifestations and therapy management frequently are the result of pathophysiological, homeostatic, and therapeutic processes operating simultaneously with synergistic and antagonistic interactions. Reasoning on this level fits well with Kuipers' mechanistic view of medical physiology as processes. I believe that there is need for both perspectives and that any system that cannot represent both views of time is likely to be deficient in its temporal-reasoning capabilities. In TOPAZ, I acknowledge the importance of this dual perspective in medical problem solving by incorporating both an interval temporal model and a physiologic process model for reasoning about a patient's clinical course.

2.1.4 Complexity and Computability in Temporal Models

In previous sections, I claimed that time introduces serious complexities for an intelligent reasoning system that must represent and manipulate temporal concepts. In this section, I substantiate this claim. To support my case, I examine the ramifications of temporal issues from two perspectives—the complexity of temporal representation and the complexity of temporal reasoning.

2.1.4.1 Representational Complexity

Time-ordered observations from a dynamic system are not totally random numbers. Relationships between system elements impose constraints on the values among

related attributes. Time imposes a precedence ordering on data that does not exist in atemporal settings. Throughout this dissertation, I use the term *time-ordered data* to emphasize this key characteristic of temporal information. The imposition of precedence has a significant influence on the complexity of the reasoning task. Using simple theorems of combinatorics, I show that, for even small sets of atemporal data with only a small number of predefined possible values (such as plus or minus), the number of potential relationships among the data (the potential relation space) is large. I extend the analysis to examine the size of the potential relation space for time-ordered data. This examination demonstrates that the size of the potential relation space for time-ordered data exhibits rapid exponential growth. These simple calculations underscore the marked increase in complexity that accompanies time-ordered data analysis. The term *potential* is emphasized because any given dynamic system will realize only a very small number of these relationships. It is precisely this knowledge about temporal constraints that makes temporal representation and reasoning so critical to systems that otherwise would be unable to limit their reasoning to the much smaller space of relations actually realizable by a physical system.

For this analysis, I define the set of observations taken at the same moment in time to be a *data vector*. For simplicity, I assume that each observation in a data vector can take only one of v values. For a data vector consisting of q observations, the number of possible data vectors that could be observed is v^q . For N data vectors that are not time-ordered, the number of possible unique data vectors that could be observed is

$$N * v^q \tag{2.1}$$

For the same conditions (q observations, each taking one of v values with N data vectors), but imposing a temporal ordering on the N data vectors, the number of possible unique data vectors that could be observed is

$$(v^q)^N = v^{q*N} \tag{2.2}$$

For fixed values of q and v , as the number of data vectors increases (increasing N), Equation 2.1 increases linearly, whereas Equation 2.2 increases exponentially.

As a simple example, let $q = 10$ observations per data vector, $v = 2$ possible values per observation (for example, plus or minus only), and $N = 5$ data vectors. The total number of observations is

$$10 * 5 = 50 \quad (2.3)$$

From Equation 2.1, the upper bound for the number of unique atemporal data vectors is

$$5 * (2^{10}) = 5120 \quad (2.4)$$

For the same numbers, from Equation 2.2, the upper bound for the number of unique temporal data vectors is

$$2^{10*5} = 2^{50} = 1.12 \times 10^{15} \quad (2.5)$$

With only five data vectors, the number of relationships from Equation 2.5 far exceeds that from Equation 2.4. This result demonstrates the exponential growth in relationships that occurs in an exhaustive, uninformed search of all possible relationships among time-ordered observations. The addition of a second dimension (time) quickly dominates the space of possible relationships. Clearly, an exhaustive search is intractable. Most observations are not related in a meaningful manner. These impossible or nonsensical relations should be removed from the search space. Only a small percentage of the *potential* relationships even have describable meaning, such as “increasing,” “decreasing,” or “steady.”

2.1.4.2 Inferential Complexity

Vilain and Kautz published a key study on the computation-theoretic implications of interval- and point-based temporal-reasoning schemes [127]. Two definitions are needed to understand their conclusions: (1) an algorithm that is *sound* never concludes an invalid consequence to a set of assertions, and (2) an algorithm that is *incomplete* fails to derive all the conclusions that are possible from a set of assertions. Using Allen’s interval calculus, Vilain and Kautz reached the following conclusions about the complexity and tractability of Allen’s interval temporal algebra:

- Determining the consistency of a set of statements in the interval algebra is NP-hard. Determining all consequences of these statements (the transitive closure) also is NP-hard. These findings imply that no polynomial-time procedure exists that can decide

Interval Notation	Point Notation	Graphic Notation
$A - (< mo) - B$	$A - -(<) - B -$ $A - -(<) - A +$ $A + -(<) - B +$ $B - -(<) - B +$	
$A - (<>) - B$	No equivalent point form	

Figure 2.4 Translation of interval relations into point relations.

The top example shows how we can reformulate the interval disjunction: *A is any one of BEFORE, MEETS, or OVERLAPS B* into a point-based algebra. In point-based algebras, $A-$ denotes the starting point and $A+$ denotes the finishing point for interval A . The bottom example shows an interval disjunction that has no equivalent form in point-based temporal algebras. (Adapted from [127, pg. 381])

the consistency or transitive closure of a database of temporal intervals in less than exponential time.

- Determining the consistency of a set of statements in a point-based temporal algebra is polynomial. The constraint-propagation algorithm operates in $O(n^3)$ time and $O(n^2)$ space. The algorithm presented by Vilain and Kautz is both sound and complete for computing the consistency and transitive closure of new assertions.
- A restricted form of the interval algebra can be formulated in terms of the point algebra. For this smaller interval algebra, the constraint propagation is sound, is complete, and is polynomial in time and space.

The smaller interval algebra captures all nondisjunctive relationships (containing only one relation between a pair of intervals) and many ambiguous relationships (containing more than one relation between a pair of intervals), it cannot capture all the possible disjunctive sets. Figure 2.4, taken from [127], shows examples of complex disjunctive interval relations that can and cannot be expressed in the smaller, but computable, point-based algebra.

Allen proposed an interval-based propagation algorithm that was polynomial in time[3]. His algorithm was sound but incomplete. From the analysis by Vilain and Kautz, a polynomial inferencing algorithm that is both sound and complete probably does not

exist. With these results, an implementor is faced with three choices: (1) use the complete interval algebra, on only small databases so that the exponential algorithm does not become intractable; (2) give up completeness and use the polynomial algorithm; or (3) use the smaller algebra, which has both completeness and polynomial behavior at the expense of representational or inferencing power. Because my research summarizes data that have definite time tags, I chose the third option in TNET (Chapter 4) and in TOPAZ (Chapter 7).

2.2 Computational Models of Temporal Data

We make decisions based on our beliefs about the current state of the world. I define *facts* to be observations or measurements that provide information about the true state of the world. In a dynamic environment, the facts on which we must make decisions are changing. In a dynamic system, measurements on the same object can differ at different moments in time. Database systems that are responsible for storing and retrieving time-varying information must deal with the additional responsibilities that arise from the special characteristics of temporal data.

A central concept in database design is that a database's structure should reflect the objects and relationships in the real world. In their book on data models, Tsichritzis and Lockovsky made the following claim:

For data to be useful in providing information, they need to be organized so that they can be processed effectively. Many different ways of organizing data exist, such as tables, lists, and forms. In data modeling, we try to organize data so that they represent as closely as possible the real-world situation, yet are still amenable to representation by the computers. These two requirements are often conflicting. [122, pg. 1]

A footnote in a later chapter in the same book underscores the problem:

One wonders why data models avoid treating time separately and not like any other attribute. This may be yet another shortcoming which is there mainly for historical reasons ready to be challenged with the freedom that new technology gives us. [122, pg. 231]

Even if there were no historical accidents or technological impediments, fundamental properties of temporal data would still add complex semantic issues that can not be solved easily with the addition of a timestamp to each observation. A single clinical

measurement has multiple times associated with its value; each time has a slightly different semantic interpretation that may be useful in solving difficult diagnostic or therapeutic temporal problems. For example, a patient's serum sodium has the following times associated with its value: (1) when the order for the test was written in the chart, (2) when the order was transcribed onto a laboratory request form, (3) when the request form was received by the laboratory, (4) when the blood sample was drawn, (5) when the blood sample was logged into the laboratory, (6) when the blood sample was logged into the laboratory station responsible for performing the test, (7) when a preliminary value was obtained, and (8) when a verified value was obtained. Additional times may be generated by addendum or updates that could change the final value of the serum sodium. Microbiology tests have even more temporal tags due to multiple preliminary culture results (for example, 24-hour readings, 48-hour readings). The temporal characteristics of medical data are complicated by the effect of the clinical context on the usefulness and validity of a measurement. There are two contexts to consider: (1) the set of clinical contexts that exist when an observation is obtained, and (2) the decision-making context in which an observation is pertinent. Most clinical data have temporal durations over which the value can be considered a valid measure of the current situation. Unfortunately, this critical feature is not solely a property of the attribute. For example, a hematocrit measurement usually is valid for a few days; in the presence of active arterial bleeding, however, the same measurement is useful for only a few minutes. In medical decision making, the current clinical context also can markedly change the usefulness of past observations. In cancer chemotherapy, patients frequently received alternating courses of drug combinations. To make a decision about combination A, the physician usually must review the patient's record for information about the previous courses of combination A; conversely, when the physician is considering administration of combination B, only those observations during past treatments with combination B are relevant. Past values will therefore alternate between being relevant and being irrelevant to the current decision. Without capturing the clinical contexts that existed when an observation was recorded, we find it impossible to determine which values are relevant when we have a context-specific decision-making situation in the future. Serious errors in reasoning will inevitably occur if the problem solver uses isolated numbers indiscriminately.

New database models have been proposed to capture additional temporal features of real-world data. As new mass-storage capabilities have eliminated previous restrictions on retaining past data, a rapid growth in temporal-database research has occurred [17,

82,112]. If a database-management system (DBMS) is to reflect a constantly changing real world accurately, then the database must contain explicit structures and operations that represent essential aspects of temporal events. In this section, I examine newer data models that acknowledge the special role that time plays in the storage and retrieval of time-changing data. Although each approach offers special features and operations that explicitly manipulate the temporal aspects of the stored data, I believe that current methodologies are not addressing key aspects of time-ordered medical information. For example, none of the temporal data models address the need to capture the complex concurrent contexts that surround a clinical measurement. These shortcomings have a significant effect on what can be done with stored information by either a decision maker or a decision-support system. In Chapters 4 and 7, I present novel database extensions that support context-sensitive data storage, retrieval, and interpretation.

2.2.1 Temporal Data Models

There are two general approaches to handling time in DBMS systems: (1) time is a regular attribute that has no special status or DBMS support, and (2) time is a special attribute that has unique properties that require additional capabilities to be incorporated into the DBMS. Traditional DBMS models follow the first approach. The semantics of handling temporal relationships and constraints are specifically left to the applications programs. The DBMS provides no additional procedures to support temporal operations. The newer data models I present here reject this approach. The need to perform historical queries, to represent retroactive analyses, and to support consistent error correction with temporal data argues for capturing the special nature of time in the internal operations of the DBMS. Temporal data models are formal approaches to delineating what additional operations are required to capture the essential features of temporal data storage and retrieval.

Research on temporal data models has progressed along two divergent tracks. One body of research is attempting to extend traditional database models with new constructs and operators for additional temporal support.³ A second approach is the development of new data models that are not based on existing data models or implementations. The advantage of extending traditional models is that these models have well-studied properties and functioning implementations for developing proposed extensions. The disadvantage of

³Traditional data models are the CODASYL network model, the relational model, and, more recently, the entity-relation (ER) model.

this approach is that the extensions are added “on top of” the assumptions and constraints of the underlying data model. Creating novel data models avoids these restrictions, but the properties of such models are less well understood. In addition, building prototype implementations to test various hypotheses about the capabilities of the new data model requires considerable work. Both approaches have resulted in interesting models and implementations. I shall examine representative systems from both lines of research. This review is not exhaustive; the extensive bibliographies by Bolour [17] and McKenzie [82], and the review of ongoing projects by Snodgrass [112], provide a detail survey this field.

2.2.1.1 Temporal Extensions to Traditional Data Models

Conventional DBMSs do not explicitly record the time-varying aspects of the real world. Numerous schemes have been proposed to support temporal information processing by incorporating one or more time attributes directly in the data model. Actual implementations have incorporated special operations for manipulating these additional time attributes. Snodgrass and Ahn [115] defined three temporal dimensions—*transaction time*, *valid time*, and *user-defined time*—to distinguish among various approaches to modeling temporal data. *Transaction time* is the time at which a piece of new information is placed into the database and is made available for use and review by other users. *Valid time* is the time when the new piece of information became valid or true. Because there usually is some time lag between when information become known and when it is recorded in a DBMS, these time dimensions usually are different. *User-defined time* is any other time tag that is not captured by the semantics of *transaction time* or *valid time*. Snodgrass considers *user-defined time* to require no special DBMS support other than an internal representation and simple input-output facilities. Unfortunately, medical data fit poorly into the Snodgrass scheme, because most of the potential time tags are neither *valid time* nor *transaction time*, and therefore would fall into the unsupported *user-defined time* category.

Based on the distinction between *transaction time* and *valid time*, Snodgrass and Ahn have proposed a taxonomy of extensions to traditional data models that attempts to capture different temporal aspects of the real world (Table 2.2).

Table 2.2 A taxonomy of traditional data-model extensions.

Name	Transaction time	Valid time
Snapshot		
Rollback	x	
Historical		x
Temporal	x	x

2.2.1.1.1 Snapshot Databases

Snapshot databases represent the state of the world at a particular moment in time. Neither *transaction time* nor *valid time* is explicitly represented in a snapshot database. The database model of reality is changed immediately on the insertion of new data into the DBMS. During the interval of time between the moment when new information is known and that when it is entered into the database, a snapshot DBMS will be out of date with respect to the current reality. After new data have been inserted, the past database state is lost. This simple modeling perspective is typical of database implementations that do not have time as an attribute or a key. Without any special temporal structures and operations, in most cases, this model is too simplistic to be useful in supporting complex data retrieval in dynamic settings.

2.2.1.1.2 Rollback Databases

Rollback databases augment the storage and retrieval of time-varying data by adding a time tag that identifies when a new value was entered in the DBMS. In the Snodgrass and Ahn taxonomy, this time tag is the *transaction time*, because all new values are associated with the time each item was entered in the database. The term *rollback* refers to the ability to return the database to a previous state using the transaction-time tag. Rollback databases do not attempt to model reality, because the entry of a datum in the DBMS may occur long after the time when that datum's value was known.

Ariav proposed a data model for rollback databases called Modal Data Management (MDM), which was an extension to the Dynamic Alerting Transaction Analysis (DATA) system developed at the University of Pennsylvania [8].⁴ All transactions in MDM

⁴MDM should not be confused with a later model proposed by Ariav called the Temporally Oriented

were appended to the previous database state. Additions, modifications, and deletions of database entities were nondestructive, cumulative additions to the database. Two new access routines explicitly manipulated the *transaction time* stamp associated with every database transaction. SETTIME allowed users to view the database as it existed at some previous time. Entities that may have been deleted were still available for examination if the time of SETTIME was earlier than the time of deletion for those entities. ROLLBACK allowed the user to manipulate the database as it existed at an earlier time. The state of the database world was rolled back to its state at the earlier time. After a ROLLBACK, the user was able to perform new DBMS operations on the rolled-back state of the database.

Rollback databases have properties that distinguish them from other data models. Because all database operations are nondestructive appends to the current state of the database, rollback databases do not support retroactive or postactive changes. Similarly, errors in past tuples can be neither corrected nor forgotten, although an error can be overridden if it occurs in the current database state.

2.2.1.1.3 Historical Databases

Like rollback systems, historical databases augment the storage and retrieval of data by adding a new time tag. In this case, however, the time tag identifies when a value or change existed in the real world, rather than when it was entered in the database. This time tag is the *valid time* property of Snodgrass and Ahn. Historical databases model changes in the real world as best these changes are known. Various models of historical databases have been proposed, such as CSL [18], TERM [58], IL_s [24] and HRDM [25]. Competing models differ in the level of representation of the temporal dimension, which may be the relation level, the tuple level, or the attribute level. Each level of temporal modeling has significant storage and retrieval differences, although more recent work has concentrated on the storage-intensive attribute level.

The Historical Relational Data Model (HRDM), proposed by Clifford and Crocker, is a historical database that is an extension of the relational model [25]. HRDM associates a time tag with each attribute. The intuitive notions in HRDM are based on the analogy of an attribute *lifespan*. An attribute is *born* when the first record of any information about it enters the database. The attribute may subsequently *die* and *reincarnate* when it ceases

Data Model (TODM), which described a temporal database model.

and resumes to be modeled in the database over time.

HRDM defines three attribute types: (1) constant attributes (CA), which are atemporal; (2) time-varying attributes (TVA), which vary over time; and (3) temporal attributes (TA), which are simple time measurements. TVAs appear in historical databases only. They represent the change in an attribute over the lifespan modeled in the database. TVAs are functions that map from the time domain into a value domain. Because attributes can die and then reincarnate over the database's lifespan, HRDM defines three distinct null values: (1) the value of a TVA is *unknown* at this point in time, (2) the value of a TVA is *nonexistent* at this point in time, and (3) the value of a TVA has *never been known* at any point in time. Researchers currently are attempting to develop consistent temporal extensions to the relational operators PROJECT, SELECT, and JOIN. The key problem is defining an extended algebra that is closed for CA, TVA, and TA attributes.⁵ Operators in a closed historical algebra take historical relations and return historical relations that can be applied to subsequent operators. Because CAs and TAs are values and TVAs are functions from time to value, the new algebra must define consistent interpretations of performing a PROJECT, SELECT, or JOIN operation on attributes with different temporal properties.

The selection operator, $\sigma_{\mathcal{F}}(R)$,⁶ is sufficiently complex to demonstrate the problem caused by TVA attributes. Suppose the selection formula $\mathcal{F}$ contains an operand i where the corresponding i th component in $R(t)$ is a TVA attribute. The value of the TVA given in the selection formula may be a specific time (such as *1 July 1988*), a range or interval of time (such as *from 1 January 1988 to 1 July 198*), or the entire lifespan of an attribute. For specific times, the database may or may not have a value recorded on that date. If no value is available at the specified time, an interpolation function must determine a value for that date. The interpolation function may be a complex attribute-specific temporal function, or it may be simply the *continuity assumption*, which returns the last recorded value. When alternative selection formulae are combined with the standard atemporal selection, there are seven variations to consider. The semantics for each combination must be precisely defined to be logically correct, algebraically closed, and intuitively reasonable. This example shows the additional complexity that temporal attributes add to the SELECT operator. The

⁵TA attributes are essentially atemporal, although they take values from the time domain. For defining an extended relational algebra, TAs and CAs have the same property of remaining constant over time, whereas TVAs are functions from times to values.

⁶The complete definition of the selection operator in the relational calculus is the set of tuples such that $\{t|R(t) \wedge \mathcal{F}'\}$, where $\mathcal{F}'$ is the formula $\mathcal{F}$ with each operand i , denoting the i th component, replaced by $t[i]$. From [124, pg. 158].

problem of defining reasonable semantics to the more complex JOIN operator is even more difficult. This is an area of active research; no single set of definitions has been endorsed yet by the database community.

Historical databases have important characteristics that are unique to this class of data model. Changes stored in a historical relation represent changes in the real-world entity. As errors or new information about past changes are realized, the historical relation can be changed to correct the erroneous or incomplete historical record. In this way, a historical database always models the past as best as that past is known. Unlike in a rollback database, these changes to the past history are destructive; it is not possible to reconstruct the previously erroneous history once the updated information has modified the historical record. Thus, although a previous historical relation may have contained an error, once that error is detected and corrected, the database once again models reality. However, the database can no longer reconstruct the erroneous state that existed prior to the correction. This is a key distinction between historical and rollback databases: Rollback databases can roll back to an incorrect previous snapshot, whereas historical databases can represent the best current knowledge about the past [114].

2.2.1.1.4 Temporal Databases

A temporal database contains *transaction-time* tags, *valid-time* tags, and new data-manipulation operators for these special attributes. A temporal database embodies the added temporal features of both the rollback and historical data models. Like a rollback database, a temporal database maintains changes as nondestructive additions; in this case, however, the additions are added to historical relations. Because a temporal database is a sequence of historical databases, a rollback operation returns the historical database that existed at the rollback time. The resulting historical database represents the best state of knowledge about the real-world entities at that time. It may contain errors or incomplete data that were updated later, but these errors are maintained if the rollback time is before the time when the errors were corrected. As is true of historical databases, at any given time, the state of the temporal database reflects the best state of current knowledge about all changes in all entities and attributes in the real world.

TQUEL is an extension to the Ingres relational-database system [118,116] that implements a temporal database by adding both *transaction-time* and *valid-time* attributes.

It is an example of a temporal database implemented on top of a preexisting atemporal data model. To emulate a temporal database, TQUEL augments each tuple with two *transaction-time* attributes and two *valid-time* attributes. Time is represented as a 32-bit internal number, with a distinguished value representing “forever.” The TQUEL research has emphasized the formulation of extensions to traditional relational query languages needed to support temporal database operations. Therefore, I shall examine TQUEL in the context of temporal query languages, in Section 2.2.2.2.

2.2.1.2 Novel Data Models

The storage and retrieval of time-sensitive data has been a concern of computer scientists outside of the traditional database community. AI researchers have long recognized that time plays a significant role from both an inferencing and a database perspective. The AI community has used symbolic-programming techniques rather than the models and methods that were evolving simultaneously in the database community. Thus, the temporal databases motivated by the needs of symbolic problem solving have a flavor and style different from those of the more formal methods developed by the database community. In this section, I shall discuss a few key systems derived from this line of research that have concentrated on storing and retrieving time-varying facts about a changing world.

2.2.1.2.1 Network-Based Temporal Databases

Networks are a collection of *nodes* and *arcs*. Nodes represent objects, concepts, or situations in the domain; arcs represent relations between pairs of nodes [9]. Using this approach, nodes can be used to store facts about the world at some moment, and arcs can be used to capture the temporal relationships between groups of facts.

Kenneth Kahn used networks to build a domain-independent temporal database-management subsystem, called the TIME SPECIALIST, to support general problem-solving systems that must represent temporal concepts [55,56]. The TIME SPECIALIST was initially envisioned as a support subsystem for storing and answering questions about the temporal features of a patient’s clinical course for the Present Illness Program [99], although the connection of these two programs was never fully accomplished.

The TIME SPECIALIST used three methods to store new temporal information (Figure 2.5). The simplest and fastest method placed a new fact into a chronologically

sorted date line. Facts that had explicit date references (for example, 1 July 1988) were stored in this manner. Dates that were not fully specified (for example, July 1988) were stored in the date line with upper and lower bounds. A second storage method associated a new fact with a SPECIAL REFERENCE EVENT. A SPECIAL REFERENCE EVENT was any event that was well known to the system; that is, any event that had an explicit date reference. Thus, a statement such as "a few days after my last birthday" would be associated with the reference event representing the explicit date of my last birthday. The third storage method was BEFORE-AFTER CHAINS, which allowed the TIME SPECIALIST to maintain a sequential ordering on a series of related events. BEFORE-AFTER CHAINS maintained only the temporal ordering of events and therefore required the least temporal information to store a fact into the database. For dates without fully specified times, temporal imprecision or uncertainty was represented by FUZZ EXPRESSIONS, which were used to calculate upper and lower bounds on the time of a fact relative to previous statements in the database. Expressions such as "a few weeks," "shortly after," and "two or three years ago" were translated into dates with various fuzz factors. Simple interval addition was used to calculate relationships between two events having FUZZ EXPRESSIONS. New facts were checked for consistency with previously stored facts about other temporal events. If an inconsistent fact was entered, the system could detect the inconsistency, print the set of facts that were causing the inconsistency, and repair the inconsistency based on instructions provided by the user.

Kahn's system could be used to answer questions such as "When did some event happen?" "Did event A occur after event B?" and "What happened during a specific interval of time?" Although more difficult questions might be handled by the representation scheme, a query language for formulating more complex queries was not constructed.

Mittal described a network-based database, called PATREC, that extended the ideas of Kahn's TIME SPECIALIST by grouping related events into EVENT CLUSTERS organized into event hierarchies [89,88,90]. An EVENT CLUSTER contained two or more events that were considered to occur at the same time, although each event within the cluster could have had slightly different actual times. Certain EVENT CLUSTERS were sufficiently important to the problem domain that they were termed KEY EVENTS. A KEY EVENT could form a hierarchy of EVENT CLUSTERS that were conceptually associated with the KEY EVENT. The set of EVENT CLUSTERS subsumed by a KEY EVENT formed an EPISODE. This event hierarchy partitioned the EVENT CLUSTERS into those

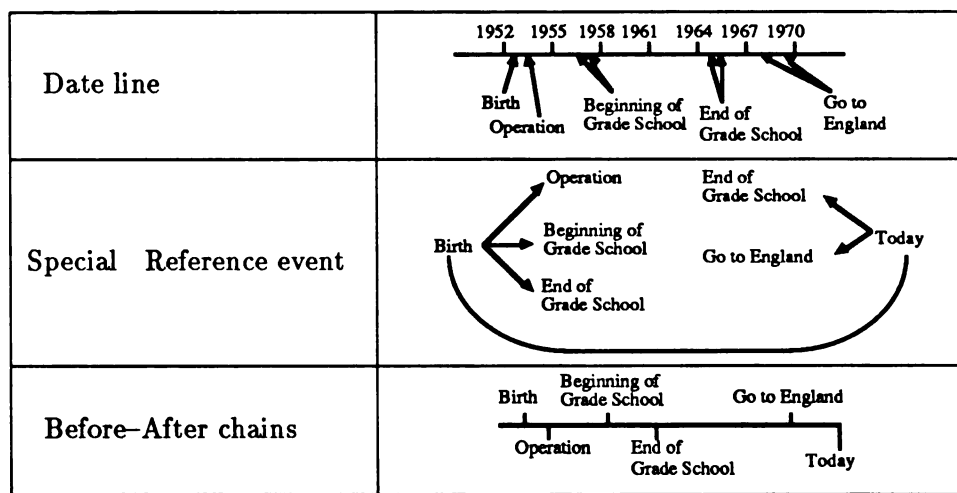


Figure 2.5 Three methods used by the TIME SPECIALIST to organize temporal statements.

that occurred before, during, and after the KEY EVENT. This organization was similar to Kahn's BEFORE-AFTER chains, extended to support hierarchies of conceptually related events. PATREC also supported a date line for fully specified times, and fuzz factors for incompletely specified times.

2.2.1.2.2 Constraint Networks

A constraint network is a collection of nodes and arcs, where an arc denotes a relationship that holds between the nodes connected by that arc. Unlike in a semantic network, the relationship between two nodes in a constraint network can change as new information (new nodes or additional information about arcs) is obtained. Through the use of constraint-propagation algorithms, additional information can cause new constraints (additional arcs) to be inferred between pairs of nodes that are only remotely related to the new observation. The temporal interval system of Allen presented in Section 2.1.1 used constraint networks to maintain consistency between pairs of temporal intervals as new intervals were added to the database. In that system, the constraint-propagation algorithm was used to calculate the transitive closure rules provided by Allen's interval calculus.

Kohane implemented a constraint-based temporal database system called TEMPORAL UTILITY PACKAGE (TUP) to represent temporal relationships derived from a patient's medical history [59,60]. New temporal assertions were made via RANGE RELA-

```

(RREL <first point specification>  <second point specification>
    <lower-bound distance>         <upper-bound distance>
    <context>)
```

Figure 2.6 RANGE RELATIONS in the TEMPORAL UTILITY PACKAGE.

RANGE RELATIONS (RREL) specify the temporal relationship between two point events. The upper and lower bounds specify uncertainty about the precise time between related events. (Adapted from [59, pg. 171])

TIONS (RRELs), which describe how one event relates to another (Figure 2.6). Constraint propagation inferred new temporal bounds on any third interval that shared a point with the newly asserted fact. The newly calculated bounds could trigger additional constraint calculations. RRELs asserted new relationships between pairs of temporal point events. Intervals were represented as two RRELs that related the starting and ending points for each pair of intervals. This scheme was analogous to the restricted point-based temporal logic proposed by Vilain and Kautz (Section 2.1.4.2).

Kohane used TUP to construct a rudimentary expert system, called Temporal Hypothesis Reasoning in Patient History (THRIPHT), to explore the performance and computational properties of constraint propagation-based temporal reasoning on diagnostic problem solving. As new facts were entered into THRIPHT, new upper and lower temporal bounds were calculated by TUP on all intervals that were temporally related to the new information. In THRIPHT, no diagnostic reasoning occurred until all available historical information was entered and the TUP constraint algorithm had finished calculating the smallest possible bounds on all interval assertions in the database. THRIPHT exploited temporal constraints between facts in rule premises that searched for temporal relationships to support a possible diagnostic hypothesis (Figure 2.7). As has been the case with all network-based representation schemes, Kohane found that hierarchical techniques had to be used to keep the computational costs of network searching and updating tractable (Section 2.1.4.2). THRIPHT used a hierarchical organization derived from MDX [23]. With this hierarchical arrangement of temporal constraints, THRIPHT pursued more specific hypotheses only when initial screening rules succeeded.

```
(AND
  (BEFORE-NOW-BY-P
    ((NAME JAUNDICE) (TYPE BEGIN-INTERVAL))
    (+EPSILON) (+INFINITY))
  (BEFORE-NOW-BY-P
    ((NAME IV-DRUG-ABUSE) (TYPE END-INTERVAL))
    ((-INFINITY) (+7 MONTHS)))
  (BEFORE-NOW-BY-P
    ((NAME IV-DRUG-ABUSE) (TYPE BEGIN-INTERVAL))
    ((NAME JAUNDICE) (TYPE BEGIN-INTERVAL))
    (+2 MONTHS) (+INFINITY))))
```

Figure 2.7 Temporal rule predicate in THRIPHT.

This figure shows the predicate portion of a THRIPHT rule corresponding to the statement: "Consider acute hepatitis B secondary to intravenous drug abuse if jaundice has been manifested, intravenous drug abuse did not end before the previous 7 months, and the onset of drug abuse preceded the onset of jaundice by at least 2 months." The temporal relationships are tested using a database constructed in TUP. (Adapted from [60, pg. 83.])

2.2.2 Temporal Query Languages

A DBMS provides routines for both the storage and the retrieval of information. The user requests pertinent information by formulating a query in a formal query language. This language must be powerful enough to provide constructs for the user to express his data requirements, yet simultaneously must be restrictive enough to yield unambiguous semantics and correct answers. When a database is automatically accessed by an expert system, the database-query language determines which concepts can be formulated and queried by the inferencing system. In rule-based expert systems, the query language determines which data-retrieval concepts are expressible in rule predicates. In this section, I shall examine current research on defining specialized query languages that support the retrieval of time-varying data. In Section 4.4, I shall present the temporal query language TQUERY that I developed to meet the temporal database-querying requirements in ONCOCIN, a rule-based expert system that provides advice on cancer-therapy planning.

2.2.2.1 HRDM

Temporal attributes cause an endless array of new problems in defining consistent and intuitive query languages. In HRDM, Clifford notes the need for three distinct NULL values (Section 2.2.1.1.3). These distinctions in NULL values are directly attributable to the additional semantics caused by the existence of time-varying attributes in a historical database. Another major source of difficulty is determining the correct interpolated value when a temporal query requests a value between two recorded measurements. Some attributes have step-interpolation functions; that is, they remain unchanged until a new value is recorded. This interpolation function is the database equivalent to the *frame assumption* in the AI community. Other attributes have no interpolation function. For example, a temporal database that stores the number of parking citations given each year cannot determine the number of parking citations that were given during a particular month.

Current research has focused on defining a proper meaning of the JOIN relational operator. This definition task is complicated by the number of possibilities that may occur among attributes in a temporal database that may be referenced in the JOIN operation. Clifford notes five possibilities [26]:

1. The two relations have no attributes in common

2. The two relations share time-constant attributes only
3. The two relations share time-varying attributes only
4. The two relations share both time-constant and time-varying attributes
5. The two relations share both time-varying and timed (dated) attributes

For each of these five possibilities, the two relations may have different lifespans. Thus, if relation R1 has a lifespan from t1 to t2, and relation R2 has a lifespan from t3 to t4, it is unclear whether the relation that is created after a JOIN operation should be the intersection (possibly empty) or the union of the two lifespans. Researchers working on HRDM are attempting to provide meaningful definitions for temporal SELECT and temporal JOIN operators. As mentioned in Section 2.2.1.1.3, there currently exists no widely accepted answers to these questions.

2.2.2.2 TQUEL

TQUEL is a superset of QUEL, the atemporal relational-query language of INGRES [118,116]. The main querying operation in QUEL is the RETRIEVE statement. TQUEL adds three new clauses to this operator [1,113]:

1. WHEN is a predicate that specifies temporal relationships among participating tuples. It is the temporal equivalent to the atemporal WHERE predicate in QUEL.
2. VALID is a clause that specifies how the implicit time attributes (either *transaction time* or *valid time*) are computed for result tuples.
3. AS OF is a clause that specifies which rollback version of the temporal database to use for subsequent RETRIEVE operations.

To emulate fully the append-only requirements of a temporal database, TQUEL modifies the destructive actions of the QUEL commands APPEND, DELETE, REPLACE, and CREATE. New temporal operators include START OF, END OF, PRECEDE, OVERLAP, and EXTEND.

The WHEN clause is the major new predicate in TQUEL that specifies complex temporal conditions for tuples participating in the retrieval operation. TQUEL distinguishes

```
RETRIEVE (h.id, h.seq, i.id, i.seq)
      VALID FROM START OF (h OVERLAP i)
      TO END OF (h EXTEND i)
WHERE h.id = 500 AND
      i.amount = 73700
WHEN h OVERLAP i
AS OF "1981"
```

Figure 2.8 A representative simple TQUEL query.

This query requests information from the state of the database AS OF 1981; it requires that the two tuples coexisted at some moment. The WHEN clause is standard QUEL. (Adapted from [1, pg. 97].)

between *event relations* and *interval relations*. Event relations contain tuples that represent instantaneous occurrences; these tuples have a single *valid-time* stamp. Interval relations contain tuples that represent events that have duration; these tuples have two *valid-time* values for a starting and ending valid time. Some TQUEL constructs distinguish between the two relation types; others have default coercion rules. Event relations can be coerced into interval relations that have the same starting and ending valid times. Interval relations can be manipulated as event relations by using the BEGIN OF and END OF operators. These functions return the starting or ending valid time of an interval tuple. Relative temporal relationships between pairs of tuples are specified using the PRECEDE, OVERLAP, and EQUAL operators between two (point) events. These operators are called *temporal-predicate operators* (Figure 2.9).

We can construct complex TQUEL queries by using atemporal logical predicates, such as AND, OR, and NOT. An E-EXPRESSION contains tuple variables, temporal constants, and temporal operators that evaluate to an *event* (a time point); an I-EXPRESSION contains the same elements but evaluates to an *interval* (a starting- and stopping-time pair). A TEMPORAL PREDICATE is an expression containing any number of logical predicates operating on temporal-predicate operators that, in turn, operate on pairs of E-EXPRESSIONS or I-EXPRESSIONS.

```
RANGE OF f1 IS faculty
RANGE OF f2 IS faculty
RANGE OF a IS Associates
RETRIEVE INTO Stars (Name =f1.Name)
      VALID FROM BEGIN OF f1 TO BEGIN OF f2
      WHERE f1.Name = f2.Name AND
            f1.Rank="Assistant" AND
            f2.Rank="Full"
      WHEN (f1 OVERLAP a) AND (f2 OVERLAP a)
```

Figure 2.9 A complex TQUEL statement with temporal-predicate operators.

This query answers the question: Who got promoted from assistant to full professor while at least one other faculty remained at the associate rank? Notice the use of BEGIN OF in the VALID clause and of the OVERLAP relation in the WHEN clause. (Adapted from [113, pg. 257].)

TQUEL predefines two distinguished time points: *now* and *forever*. These time points are used as default temporal values in queries that do not explicitly use TQUEL extensions. The defaults have been defined such that standard atemporal QUEL queries return the expected tuples in TQUEL. However, Snodgrass admits that the defaults should be used carefully—they are not guaranteed to be correct in all cases.

TQUEL is especially important because it is an extension to a well-formalized query language. Snodgrass presents the formal semantics that define the intended behavior of his extensions. In addition, TQUEL is a functioning temporal database implemented on top of a snapshot relational database. Despite the inefficiencies that this approach inevitably creates, Snodgrass and Ahn have gathered relative performance and storage statistics for temporal databases of differing sizes and for temporal queries of varying complexity [1]. Currently, no other research has been able to examine these critical, practical issues.

2.2.2.3 RX

Blum constructed a collection of time-sensitive database-access functions to support complex temporal queries in a standard time-oriented database of patient records [15] as part of the RX project (Table 2.3). In RX, Blum was interested in detecting causal relationships within collections of patient records. His methodology required looking for associations between actions and effects that were separated over time. Because most patient records contain significant numbers of missing data and long intervals between successive observations, Blum had to construct methods that handled complex interpolation problems. For example, suppose a value for a patient's urine protein was recorded on day 12 and on day 20, but that there is a need for the urine-protein value 6 days prior to the day-20 visit (day 14). There is no explicit value recorded for day 14, but the value on day 12 may provide an acceptable answer to this query. The correct response requires additional knowledge about both the query and the attribute. Using Blum's routines, the query could specify whether it is permissible for the system to search for nearby substitute values when the specified value is missing. If nearby values are permitted, the query also could specify whether the search for a substitute value should occur in both directions in time (earlier and later than the specific day requested) or in only one direction (for example, only earlier). This additional information in a query required knowledge about the purpose of the query and therefore had to be supplied either by the user or by some intelligent system.

Knowledge about the attribute provided time bounds on how long a measurement remains valid. Some clinical measurements do not change rapidly, whereas others are volatile. The day-12 measurement may be a perfectly acceptable answer to the request for the day-14 value, provided that the attribute is known to be relatively stable. Unfortunately, the stability of a particular attribute may change depending on the clinical context. To resolve the interpolation issues, Blum required all queries to specify the acceptable time window during which a measurement could be used to answer a temporal query.

In Chapter 4, I present a new database model, called TNET, for storing context-sensitive temporal data. TNET borrows features from both the traditional and the novel data models. For example, like HRDM, which is an extension of the relational model, TNET maintains historical information at the attribute level. Like the TIME SPECIALIST and PATREC, which are network-based data models, TNET maintains temporal information in a hierarchy of related intervals. In Chapter 4, I shall also present a temporal query language, called TQUERY, which borrows features from both TQUEL and RX to support context-sensitive, query data retrieval. TQUERY extends the capabilities of these previous query languages by supporting queries that TNET resolves by specifying the context in which the data were obtained rather than by referring to an absolute time line or calendar date.

2.3 Computational Models of Temporal Decision Making

Intelligent systems are algorithms that use internal models to manipulate data. In the previous two sections, I focused on the structure of the internal model of time and time-ordered data. In this section, I shall turn to the algorithms that manipulate clinical data to generate medical decisions. The main theme in this section is the same as that in the previous two sections: The choice of a particular computational model of decision making limits the types of inferences and therefore the types of conclusions that can be derived within a decision-support program. As is the case with the previous computational models, compromises in complexity versus expressiveness reappear in the construction of intelligent inferencing algorithms. Time frequently is completely ignored or is skillfully finessed because of the additional reasoning complexity that it places on problem solving. In Section 1.1, I discussed why reasoning about time and change introduces serious problems for intelligent systems. In this section, I shall examine how implemented medical systems

Table 2.3 RX time-dependent database-access functions.

Access function	Arguments	Actions
DELAYED-ACTION	ACTION VARIABLE, DAY, ONSET DELAY, INTERPOLATION DAYS	Retrieve the value of ACTION VARIABLE that is ONSET DELAY days prior to DAY, using any value that falls within INTERPOLATION DAYS before DAY
DELAYED-EFFECT	STATE VARIABLE, DAY, ONSET DELAY, INTERPOLATION DAYS	Retrieve the value of STATE VARIABLE that is ONSET DELAY days prior to DAY, using any value that falls within INTERPOLATION DAYS before DAY
PREVIOUS (NEXT)	VARIABLE, PATIENT, DAY	Retrieve the last (next) entered value of VARIABLE previous (after) DAY
DURING	INTERVAL VARIABLE, PATIENT, DAY	Retrieve the INTERVAL VARIABLE only if it was present on DAY
RAPIDLY-TAPERED	POINT VARIABLE, PATIENT, SLOPE	Retrieve intervals in which the rate of change of POINT VARIABLE was greater than SLOPE
TIME-AVERAGE	VARIABLE, PATIENT, INTERVAL	Retrieve the "area under the curve" for VARIABLE over the duration of INTERVAL

have approached the *time problem*.

No medical decision-support program that has demonstrated any level of competency has been a true atemporal problem solver. Most programs, despite their use of widely dissimilar problem-solving methodologies, make strong temporal assumptions in order to apply essentially adynamic reasoning strategies in a dynamic world. In the following sections, temporal assumptions that have been incorporated into past implementations are examined. I have arranged the discussion from the most restrictive to the least restrictive, although the ordering is debatable. Note that the presence of a system in one category does not mean that that system does not contain assumptions discussed in another category.

2.3.1 State-Based Temporal Ignorance

Faced with a current clinical setting in which a diagnostic or therapeutic problem is to be solved, many systems assume that key temporal features of the problem domain can be captured by a set of atemporal labels. Systems that make this assumption reason about only the current state. I call this temporal assumption *state-based temporal ignorance* because the system is effectively ignorant of the temporal patterns that relate a succession of previous states to the current state of affairs. This assumption is not the same as reasoning in an atemporal problem domain, because temporal features about the real world are encoded into atemporal labels that can be used by the reasoning system.

INTERNIST-1 demonstrates the subtle difference between state-based temporal ignorance and atemporal reasoning. INTERNIST-1 is a diagnostic system that uses symbolic processing to associate patient manifestations with potential diseases [87,86]. The INTERNIST-1 diagnostic-reasoning algorithm is devoid of any temporal-reasoning capabilities. Patient signs and symptoms are encoded as patient manifestations through the use of a controlled vocabulary. Although most of the approximately 4000 patient manifestations are atemporal, the manifestations that describe temporal characteristics encode this information as distinct literals.⁷ Examples of atemporal and temporal patient manifestations from INTERNIST-1 are shown in Figure 2.10. Labels such as <CHEST PAIN SUBSTERNAL LASTING GTR THAN 20 MINUTES <S>> and <CHEST PAIN SUBSTERNAL LASTING LESS THAN 20 MINUTE <S>> make temporal distinctions by assigning the

⁷The majority of INTERNIST-1 manifestations are atemporal, despite the temporal nature of medical diagnosis. This contradiction seems to have been possible because the use of temporal information in describing disease-manifestation associations was minimized. Based on the investigators' own critique of the INTERNIST-1 performance [87], it appears this approach caused noticeable performance degradation.

-
- <ABDOMEN PAIN LEFT LOWER QUADRANT>
 - <ABDOMEN PAIN PERIUMBILICAL>
 - <ABDOMEN TENDERNESS GENERALIZED>
 - <ABDOMEN TENDERNESS SUPRAPUBIC>
 - <ABDOMEN TRAUMA RECENT>
 - <ABDOMEN TRAUMA REMOTE HX>
 - <CHEST PAIN SUBSTERNAL LASTER GTR THAN 20 MINUTES <S>>
 - <CHEST PAIN SUBSTERNAL LASTING LESS THAN 20 MINUTES <S>>
-

Figure 2.10 Atemporal and temporal INTERNIST-1 patient manifestations.

The first two manifestations capture geometric distinctions; the third and fourth manifestations distinguish patient symptoms from signs; the last four manifestations capture temporal distinctions.

finding to distinct labels. The program is entirely ignorant of the fact that these two labels are closely related or that the presence of one of these findings suggests future clinical events. This additional temporal knowledge is not encoded in either the knowledge-representation scheme or the diagnostic-reasoning algorithm. The presence of unique labels means that INTERNIST-1 can exploit differences in labels even if it does not know that the labels imply temporal distinctions.

2.3.2 Causal-Based Temporal Ignorance

Causal relationships tie events at one moment in time to events at later moments. Although causal and temporal knowledge are closely related, these two concepts are not identical. Causal knowledge imposes an ordering in time on related events but need not specify the actual evolution of events. Patients with identical causal events generally have similar but not identical temporal courses. Numerous medical decision-support systems, using markedly different representation techniques, have used causal information (such as

Formalism Encoding Causality	Representative Formulae
Logical constraint	$X \wedge \text{causes}(X, Y) \rightarrow Y$
Conditional probability	$P(X) * P(Y/X) \rightarrow P(Y)$
Markov transition probability	$X \xrightarrow{p} Y$
Functional relations	$Y = F(X) ; Y = M^+ X$

Figure 2.11 Formalisms for encoding causal knowledge.

Formalisms for representing causal information need not encode temporal information. Quantitative relationships that have time as an independent variable are the usual exception; they are discussed in Section 2.3.4. (Adapted from [65, pg. 54].)

physiological causal knowledge) to focus or limit reasoning. Table 2.11 outlines the most common approaches to representing causal relationships. Not one of the formalisms requires explicit temporal knowledge to capture the ordering constraints imposed by causal relationships. The systems that use causal knowledge exploit only the ordering implications of causal knowledge; they do not use temporal information in their problem-solving methodologies [97,98,27,106,120]. I call this *causal-based temporal ignorance* because these systems reason with causal knowledge, without having any explicit temporal knowledge.

ABEL was a consultation system for electrolyte and acid-base disturbances that demonstrates the use of causal-based temporal ignorance. ABEL encoded causal knowledge using complex semantic networks. ABEL permitted both deep and shallow causal reasoning because it segregated medical knowledge into five levels of abstraction, ranging from the least specific syndromic level to the most specific physiologic level (Figure 2.12). A concept represented as a single node at one level was elaborated in more detail as a network of more primitive concepts (nodes) at the next level. ABEL represented causal relations as general constraints between the attributes of a cause and the attributes of an effect. Qualitative and quantitative constraints could be used to encode cause-effect relationships. ABEL's representation scheme did not encode the temporal evolution of related events. For example, although ABEL could determine that severe diarrhea causes hypokalemic metabolic acidosis, it was unable to model the evolution of these findings over time. Metabolic acidosis can also occur as a result of methanol poisoning, but the temporal course of acidosis from this cause is so markedly different that the two diagnoses are easily separated by the temporal evolution of a patient's illness. Because ABEL encodes causal but not temporal knowledge,

ABEL would not have distinguished these two causes of acidosis from temporal differences.

2.3.3 Discrete Temporal Models

The systems I shall discuss in this section and in Section 2.3.4 have some form of explicit temporal knowledge. The categories are distinguished by the methods used to conceptualize the role of time in the decision-making task.

Discrete temporal models represent temporal relationships as abrupt changes between states. This approach was one of the earliest models adopted by AI researchers in the form of the situation calculus (Section 1.1). As demonstrated in Section 2.3.3.3, numeric temporal models also have exploited this representational paradigm. A change in state represents a new context in which prior assumptions or knowledge may no longer apply, and in which new assumptions or knowledge may now hold. For example, in the situation calculus, a change in state results in the removal of logical statements that are no longer applicable, and in the assertion of new logical statements that are permitted in the new state.

2.3.3.1 VM

VM, a rule-based ventilator-management program developed by Fagan, was one of the first symbolic-based decision-support programs to represent temporal knowledge explicitly [39,40]. Unlike the situation calculus, VM used IF-THEN rules to represent knowledge in a given state. VM used a state-based temporal model to encode changes in parameter values and expectations over time. Transitions between states were explicitly represented in a state-transition network (Figure 2.13). Within a given state, state-specific rule sets encoded the knowledge that could be used in that setting. These rules mapped a patient's observations into symbolic ranges, such as ACCEPTABLE or IDEAL. The definitions of these symbolic ranges varied in different states. For example, two states in VM were called T-PIECE and VOLUME.⁸ In T-PIECE, the patient's heart rate was considered ACCEPTABLE if it was between 60 and 120 beats per minute (bpm). In VOLUME, the ACCEPTABLE range was between 60 and 110 bpm. A heart rate of 115 bpm would be ACCEPTABLE in T-PIECE and would be HIGH in VOLUME. As a patient's state changed, new definitions for mapping from observations to symbolic interpretations were used. TOPAZ borrows from

⁸The names T-PIECE and VOLUME refer to two different modes of ventilatory assistance.

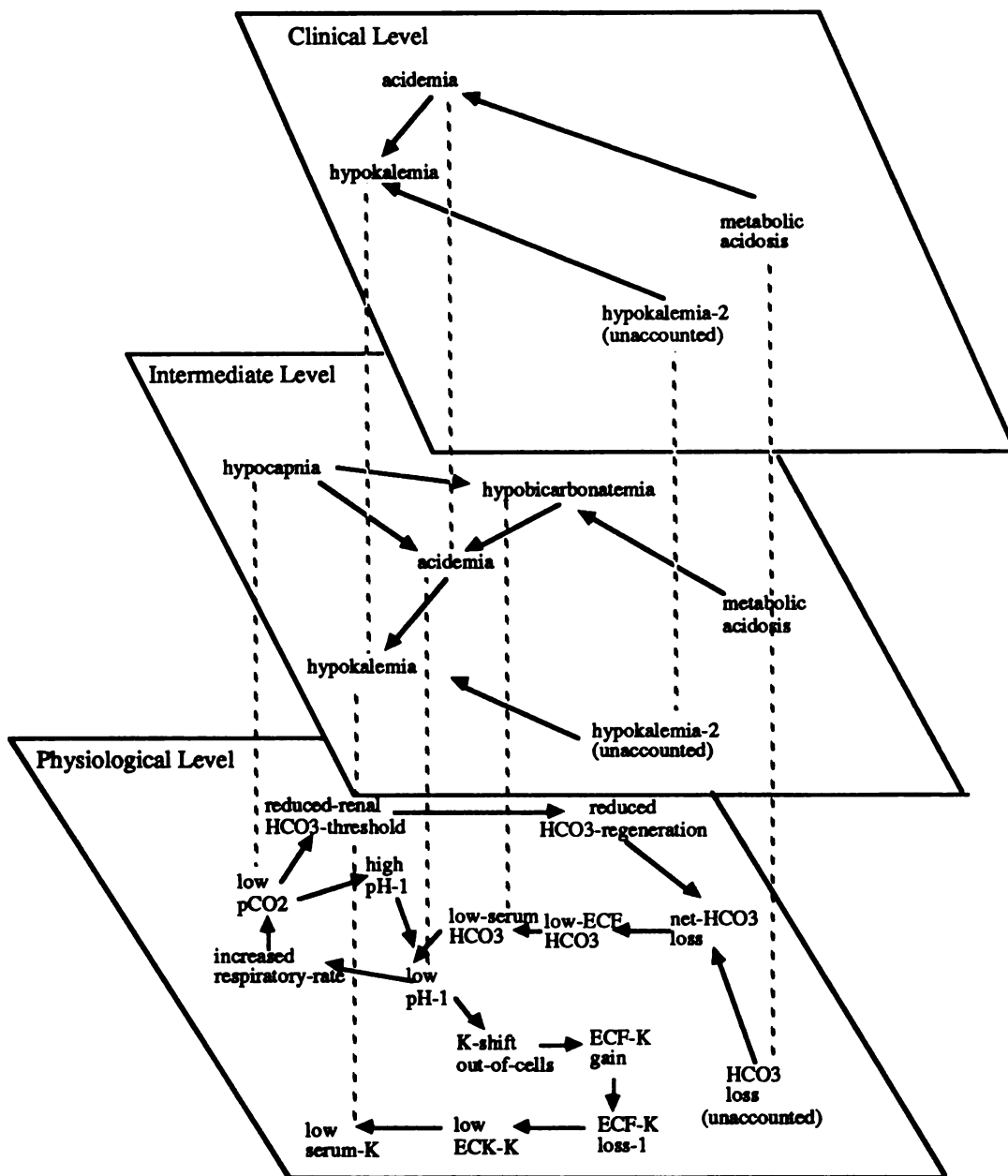


Figure 2.12 Causal abstraction hierarchy in ABEL.

ABEL represented causal clinical concepts at multiple level of abstractions. Each level was either an elaboration or an abstraction of an adjacent level. Related concepts in neighboring levels were connected by links, permitting ABEL to reason at different abstraction levels.

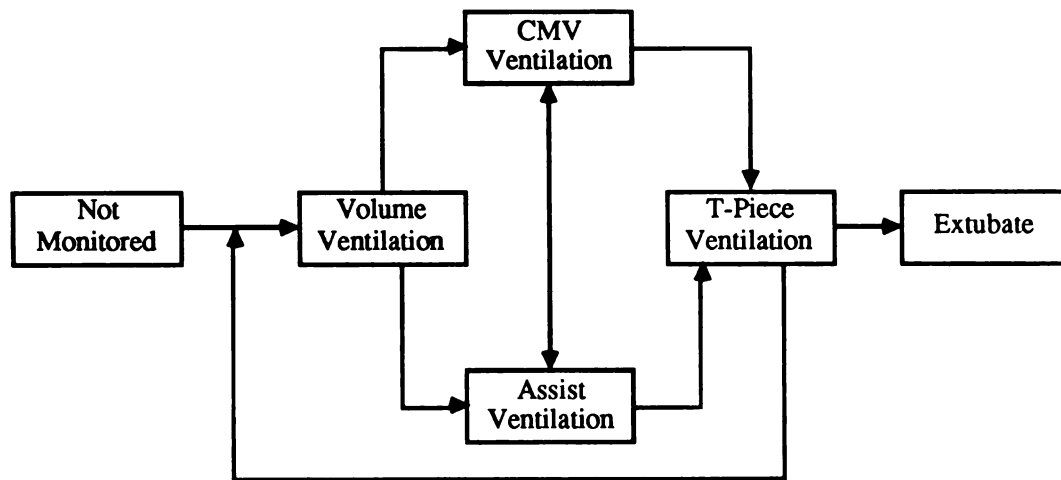


Figure 2.13 The state-transition graph from the VM system.

VM explicitly represented state changes as transitions between nodes in such a state-transition graph. Traversal of a graph node caused previous system definitions and expectations to be replaced by new definitions and expectations that applied to the new state. (Adapted from [39, pg. 79].)

VM the technique of changing rules as the reasoning context changes.

Although a state change could change the specific definitions used to map observations into a symbolic ranges, once the conversions were performed, conclusions would be compared and combined even if the initial observations were made in different states. If possible, VM aggregated conclusions made during adjoining intervals. For example, VM combined two adjoining intervals in which HYPERTENSION was concluded even if each conclusion had been derived in different states using different measurement-to-symbol definitions.

VM's temporal model incorporated two reasoning perspectives about state transitions. First, VM determined when the patient demonstrated enough improvement that a transition to the next state was indicated. VM searched for clinical findings that indicated that the current state was tolerated well. Second, VM detected when a state transition actually had occurred. This second capability was an artifact of the medical setting in which VM was implemented. VM did not have access to the actual ventilator settings so a ventilator change could occur without VM being informed of the change. VM had a set of transition rules that detected clinical features that suggested which ventilation mode the

Status Rule:	STABLE-HEMODYNAMICS
Definition:	Defines stable hemodynamics based on blood pressures and heart rates.
Applies to:	Patients on VOLUME, CMV, ASSIST, T-PIECE
Comment:	Look at the mean arterial pressure for changes in blood pressure and at the systolic blood pressure for maximum pressures.
IF:	HEART RATE is ACCEPTABLE, PULSE RATE does NOT CHANGE by 20 bpm in 15 minutes, MEAN ARTERIAL PRESSURE is ACCEPTABLE, MEAN ARTERIAL PRESSURE does NOT CHANGE by 15 torr in 15 minutes, SYSTOLIC BLOOD PRESSURE is ACCEPTABLE,
THEN:	The HEMODYNAMICS are STABLE.

Figure 2.14 Sample rule from VM.

Because VM used state-specific rules to convert numeric observations into state-independent symbolic ranges, the remainder of VM's rules were free of state-specific conditions. Note that the predicates in this rule do not make any reference to the current or past clinical state. Capitalized words are variable names and variable symbolic values in the VM rule language.

patient was receiving currently. Without these state-transition-detection rules, VM could have used rules that did not apply in the current ventilator context.

Once VM's context-specific rules converted raw measurements into context-free symbolic ranges, the remaining VM rules ignored context changes. Thus, VM rules encoded temporal knowledge relatively free of context considerations⁹ (Figure 2.14).

2.3.3.2 TEMPORAL CONTROL STRUCTURE

TEMPORAL CONTROL STRUCTURE (TCS) is a symbolic system that supports the construction of state-based temporal models by providing a mechanism to fragment time into homogeneous intervals [74,104,105] (Figure 2.15). A *homogeneous interval* is one during which no new events start or stop. After the time line has been partitioned, the appropriate rule set can be applied over the interval without concern about the sudden appearance of a new event or context that would invalidate a premise or a conclusion. Russ terms this feature *steady-state abstraction*, because the reasoning system does not need to

⁹The major exceptions to this point were the state-transition rules, which determined whether a new ventilator setting had occurred.

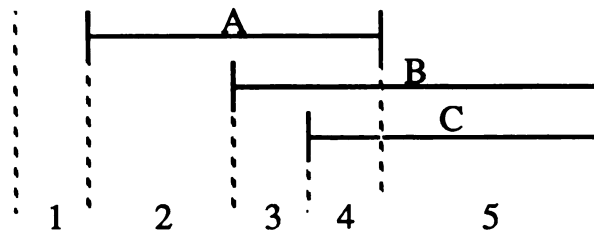


Figure 2.15 Partitioning of time into homogeneous intervals by TCS.

Three events—A, B, and C—start and stop as shown. TCS partitions the time line into five intervals during which no new event starts or stops (homogeneous intervals). Rules that execute over the duration of a homogeneous interval do not need to contend with the appearance of a new event which may validate or invalidate the rule premises or conclusions during that interval. (Adapted from [104, pg. 166].)

consider the effects of time within a TCS interval fragment. Unlike any of the previous systems, TCS has the ability to maintain context dependencies between data and conclusions. TCS exploits its knowledge about these dependencies to handle correctly data that arrive out of chronological order. If a past value is changed or a previous measurement becomes available at a later date, TCS updates current conclusions based on the newly obtained past information.

Because the start or stop of a new event causes the creation of a new TCS interval, adjoining intervals represent new states. The new event may cause a change in the set of facts or in the reasoning methods that are valid. TCS does not attempt to determine the applicability of a particular reasoning method to a change in state. To the contrary, TCS considers the reasoning methods to be conceptual black boxes. In this sense, TCS supports temporal reasoning by ensuring homogeneous intervals and by maintaining data dependencies, but TCS does not perform any temporal inferencing directly.

2.3.3.3 Markov Models of Prognosis

The systems previously discussed have mostly emphasized symbolic approaches to medical decision making. However, numeric techniques also must make temporal assumptions in order to construct computationally useful models of the problem domain.

Decision analysis is a numerical modeling technique that permits the represen-

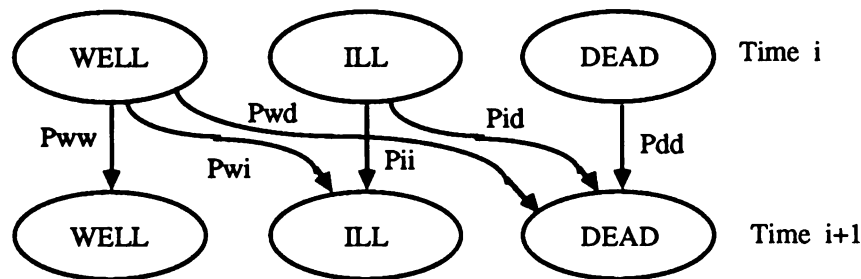


Figure 2.16 A three-state Markov model.

For each time tick, a patient has a known probability of making a transition from the current state to one of the connected states. Transition probabilities may be constants or functions over time. In either case, Markov models are discrete temporal models. (Adapted from [10, pg. 421].)

tation of chance events that can occur during a patient's clinical course. Unfortunately, standard decision analytic models can represent only those chance events that occur once over the time course of the decision. In many medical settings, chance events can occur repeatedly (such as experiencing tachycardia) or can remain a constant possibility over long period of time (such as dying from a heart attack). Beck and Pauker introduced the *Markov model of prognosis*, which is an numeric discrete temporal model that extends traditional DA methods to accommodate recurring chance events [10].

A Markov model of prognosis represents the time course of a disease as a series of discrete states. Each state has a number of possible transitions into other states. A patient's clinical course is viewed as infrequent transitions between states. For each fixed slice of time, the patient has a certain probability of moving from one state to another (Figure 2.16). Certain states are called *absorbing states*; once a patient enters an absorbing states (for example, DEAD or CURED), no additional transitions to other states can occur. Although general Markov models permit transition probabilities that are functions over time, Beck's models assume constant transition probability for simplicity and for computational efficiency.

By segmenting the clinical course into unique disjoint states, Markov models conceptually represent discrete temporal models. By embedding the evolution of a patient's clinical course as a Markov model into a larger decision analytic model, the physician can capture state-dependent patient and physician preferences. State-dependent preference are

conceptually similar to the state-specific goals and expectations encoded as rules in VM.

2.3.4 Continuous Temporal Models

Continuous temporal models generally employ numeric formulae, consisting of smooth functions with time as an independent variable. As noted in Section 1.1, however, models based on numeric formulae still contain assumptions that justify either a particular equation or a particular form of an equation. A change in state may result in the need for a change in formulae. For example, a change from an equilibrium state to a nonequilibrium state will invalidate equations based on steady-state assumptions. Within a stable state, however, continuous temporal models provide a measure of continuous reasoning that is not present in simple state-based methods.

2.3.4.1 DIGITALIS THERAPY ADVISOR

Decision-support systems for drug therapy frequently exploit numeric models to express pharmacokinetic knowledge about the temporal characteristics of the absorption, storage, and excretion of a drug. The DIGITALIS THERAPY ADVISOR combined both symbolic and pharmacokinetic methods to derive dosage recommendations for patients receiving digitalis therapy [110,119]. The final recommendations were based on a numeric model of digitalis metabolism. The pharmacokinetic model was used to calculate an initial loading dose, adjusted for the clinical indications for giving digitalis. Thus, for patients with specific medical conditions, the DIGITALIS THERAPY ADVISOR would recommend a low loading dose and would propose a slow schedule for reaching therapeutic serum drug levels, whereas for patients with more serious conditions, the program would suggest a markedly different dosing schedule. The symbolic-reasoning portion was used extensively during follow-up consultations, during which the program had to assess the therapeutic and toxic conditions that might have appeared since the previous evaluation. The symbolic-reasoning portion of the DIGITALIS THERAPY ADVISOR could change model variables or desired predicted levels if a particular clinical or toxic condition was present. In this way, the symbolic-reasoning component ensured that the pharmacokinetic model was consulted only if the current context permitted more digitalis therapy. In addition, the symbolic-reasoning system altered model parameters and goal states such that the model predictions were consistent with the current therapeutic context.

2.3.4.2 HEART FAILURE PROJECT

The cardiovascular system is highly dynamic. Sophisticated cardiovascular reasoning requires complex temporal and causal reasoning. The HEART FAILURE PROJECT (HFP) contains a detailed model of the cardiovascular system for diagnostic and therapeutic reasoning about cardiac failure [72,73]. In HFP, dynamic cardiac relationships are represented as a qualitative causal network relating physiological, therapeutic, and causal parameters (Figure 2.17). The system uses constraint-satisfaction algorithms to ensure the logical consistency among variables in the causal network, as well as mathematical techniques to propagate the quantitative effects of a perturbation throughout the cardiovascular model. For temporal reasoning, HFP makes three key assumptions: (1) the cardiovascular system is initially at steady state, (2) following clinically reasonable perturbations, the inherent stability of the cardiovascular system will cause that system to settle at a new steady state, and (3) the dynamics of changing from one steady state to another can be approximated by piece-wise linear changes between states.

In HFP, techniques from signal-flow analysis were adapted for use in propagating changes in parameter values in response to clinically reasonable perturbations from steady state. The modified technique supports feedback loops, which are a significant feature of the cardiovascular system and an important feature of normal homeostatic processes. The continuous temporal model in HFP is encoded not as a set of formulae with time as an explicit independent variable, but rather as a causal propagation network, with time relating process effects ("signals") between connected nodes. If the program is given knowledge about the hemodynamic effects of standard cardiovascular drugs, it is able to predict correctly the cardiovascular implications of a simulated dose of drug by propagating the therapeutic effects through the network. Because HFP encodes a continuous temporal model, the simulation can be stopped before the model has reached the new steady state. Thus, unlike most of the systems previously discussed, HFP can provide intermediate results as the model moves from one state to another.

2.4 Summary

My main purpose in this chapter was to examine various temporal-reasoning and temporal-representation methods that have been used to solve different temporal problems.

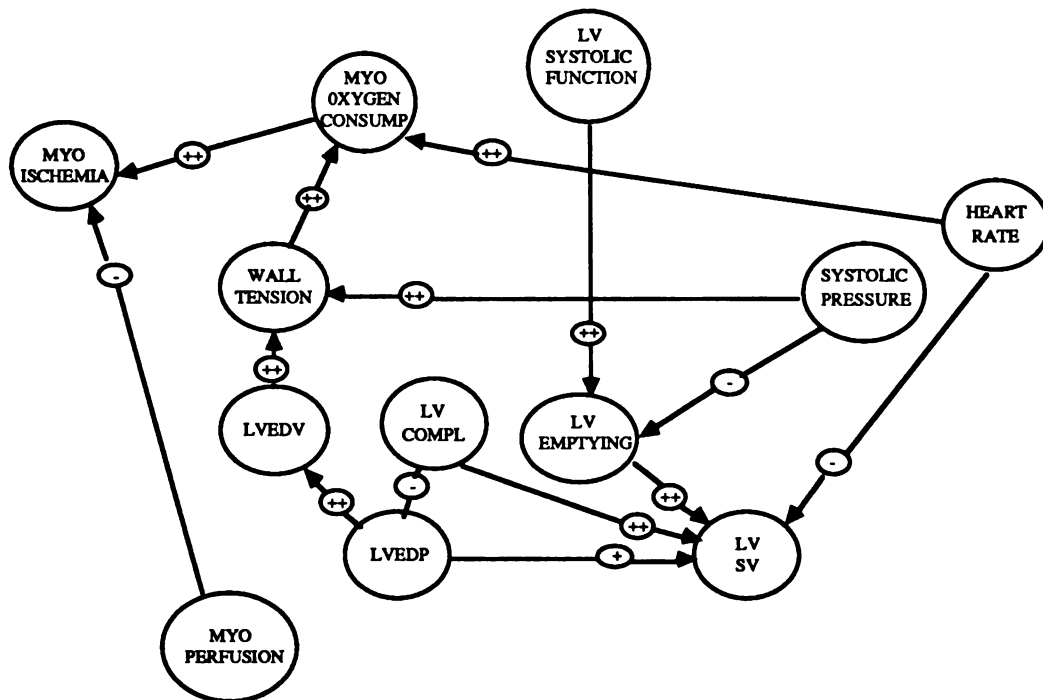


Figure 2.17 A portion of the cardiovascular-network model in HFP.

Nodes in the network represent physiological entities that affect cardiovascular hemodynamics. Links between nodes represent causal and temporal relationships between pairs of variables. All relationships are expressed as linear equations; propagation of perturbations throughout the network assumes that piece-wise linear approximations are acceptable.

Especially striking is the wide range of problems and solutions that have been examined by researchers from many disciplines. My work rests at the interface between at least two of the fields represented in this chapter—database management and artificial intelligence. TNET in ONCOCIN (Chapter 4) is motivated mostly by issues of database storage and retrieval; ETNET in TOPAZ (Chapters 5 and 7) is motivated mostly by issues of temporal representation and reasoning. In Chapter 9, I shall reexamine TNET and ETNET, and shall compare the results from these two systems to the previous work presented in this chapter. In particular, I shall examine TNET and ETNET as knowledge-representation structures (Section 2.1.1), as database-retrieval structures (Section 2.2), and as decision-support structures (Section 2.3) for representing and manipulating temporal relationships for intelligent medical decision making.

CHAPTER 3

Temporal Reasoning and the Medical Record

Recurring episodes of illness are a hallmark of many diseases. Understanding a patient's previous response to past therapy is an important task for the physician faced with making clinical decisions for a current episode that is similar to previous events. For example, a patient having a good response to current treatment is likely to remain on the same therapy, whereas another patient demonstrating progressive deterioration probably will be moved to a different therapeutic plan. In diagnostic, prognostic, and therapeutic problem solving, the chronology of disease symptoms, exacerbations, remissions, cures, and failures gives the physician a glimmer of what may lie ahead for his patient.

Clinical care generates a staggering number of data. The intensity of hospital care rapidly produces an enormous volume of data; the duration of chronic care slowly produces numerous smaller aggregates of data. For the physician faced with finding salient prognostic and therapeutic features within a patient's medical record, interpreting and summarizing this large body of information is a complex and demanding task. The information available for interpretation is often incomplete or ambiguous. Few principles guide the interpretation process; most interpretations involve extensive use of heuristics or judgmental shortcuts to fill in missing pieces. It is not surprising that experienced clinicians are more skilled at this

task than are novices [46,29].

A summary condenses information into concise statements. An *effective* summary highlights the few features that have special meaning and utility. This chapter examines the complexity of knowledge required to abstract clinically significant temporal features from a patient's medical record. I shall argue that summarizing time-ordered medical data requires knowledge about (1) the static and the temporal relationships among observations, (2) the context from which the observations were drawn, and (3) the expected evolution of observations over time. For an decision-support system to exploit temporal clinical features found in time-ordered data, these three diverse types of knowledge must be encoded in a form that can be manipulated by the program. A central hypothesis of this work is that the breadth and diversity of the knowledge required for intelligent interpretation of time-ordered medical data cannot be represented using a single formalism (see page 2). In this chapter, I shall focus on the content of this knowledge. In Chapters 4 through 8, I shall focus on representing and manipulating this complex medical temporal knowledge in intelligent reasoning systems.

3.1 The Problem Domain: Cancer Chemotherapy

In 1986, an estimated 930,000 new cases of cancer were diagnosed and 472,000 people died from cancer-related causes [109]. For cancers that cannot be cured by surgical excision, chemotherapy, radiation therapy, or combined therapies provide the only avenue for disease-free survival or cure. The search for chemical agents that could cure cancer began in the late nineteenth century when Ehrlich discovered the first alkylating agent [36]. Disappointing results with single agents resulted in the use of multidrug combinations, which heralded the era of combination chemotherapy. More effective treatment has resulted in the evolution of modern chemotherapy regimens that incorporate up to 10 agents which are delivered in complex temporal sequences.

Most cancer chemotherapies have significant toxic effects caused by unavoidable damage to normal tissues. Close patient monitoring and frequent treatment adjustments are required in aggressive cancer treatments. Actions such as drug-dose modification, drug deletion, and treatment termination are based on the patient's past reactions, current findings, and estimated prognosis. It is well known that early recognition of evolving untoward trends can greatly reduce patient morbidity. Due to the complexity and toxicity of cancer

chemotherapy, a detailed understanding of the patient's past medical record is mandatory for safe and effective medical decision making.

3.2 The Medical Record

Despite the marked increase in data gathering in modern clinical care, the structure of the medical record has changed little since the initial adoption of this form of information storage. New information is recorded and stored on separate sheets of paper. A time or date stamp denotes when the information was obtained. By default, the traditional medical record represents time as a linear sequence of paper sheets.

The medical flowsheet is an important new addition to the basic medical-record structure. It integrates patient information that usually is scattered over many pieces of paper. The flowsheet provides a unified location in which to place multiple data in proper temporal sequence (Figure 1.3). When data that previously were widely dispersed throughout the traditional medical record are placed into rigidly aligned columns in strict temporal order, relationships that might have gone undetected may become apparent. As a data-organizing tool, flowsheets have been adapted successfully for use in data-intensive, temporally sensitive clinical settings such as the intensive-care unit and oncology clinic. However, flowsheets only present data in an organized fashion; the task of interpreting the patterns embedded in the items on the flowsheet is still left to the physician.

Computer-based medical-record systems can improve the accessibility of textual clinical information[13,12,80,81,130]. Unlike the paper-based record, the electronic medical record is not constrained to a particular sequence or structure for displaying information.¹ Although access to more complete information is provided, the task of integrating and interpreting the clinical data is still left to the physician. Because electronic records increase the volume of information accessible to the physician, the task of integrating, interpreting, abstracting, and summarizing the patient's record actually becomes more difficult.

In rare clinical settings, the presence of a simple observation has high decision making value. For example, a life-threatening drug allergy is an observation that usually does not depend on other clinical variables for interpretation.² More commonly, an ob-

¹Efficiency considerations usually dictate that some internal constraints be imposed on the storage of data. These constraints may limit the way in which information can be retrieved and displayed.

²There are medical circumstances in which even this simple example is incorrect. Rarely, a serious drug allergy occurs only in conjunction with the use of a second drug or only in the presence of another clinical

servation cannot be correctly analyzed in isolation from the medical context in which the observation was made. The term *medical context* refers to all concepts that are potentially relevant to the health or illness of the patient and that could potentially change the meaning of an observation's value. There are at least three types of knowledge needed to understand a patient's current medical context : (1) knowledge about the physical entities and their relationships that may influence or modify the observations, (2) knowledge about how previous events may affect on the current interpretations and measurements, and (3) knowledge about which past and current findings are part of an evolving context that begins in the past and extends into the future. It is unlikely that the physician will ever have enough information to characterize the actual medical context completely but, by skillfully using his medical knowledge, he can make educated guesses about the current situation. The key point is that the interpretation of medical data requires the knowledge about relationships among clinical entities in the past, present, and future.

Single measurements, other than age, gender, and ethnic group, are rarely sufficient to allow the physician to draw clinical conclusions that have therapeutic consequences. Change is an important feature that cannot be captured by a single measurement. Many clinical concepts are expressed only as changes in clinical state. What knowledge must be applied to convert observed measurements and temporal changes into clinically useful therapeutic implications?

Suppose no additional knowledge was available to interpret the data plotted in Figure 3.1. What could we say about this plot? The white blood cell (WBC) count dropped 2 weeks after chemotherapy was administered, stayed down for 2 more weeks, then slowly returned to a level lower than the starting value. This is a simple *description* of the plot; it is not an *interpretation* of what the plot says about the patient. Without more knowledge, we do not know whether the drop in WBC counts was particularly precipitous, whether the nadir was especially low, whether the recovery was excessively prolonged, or whether the lower steady level after treatment was expected. More important, we do not know whether this temporal pattern implies that the patient was markedly sensitive to the myelosuppressive effects of the drug or has demonstrated irreversible bone -marrow toxicity—conclusions that would have a significant influence on future decisions to give more chemotherapy. We need to have detailed knowledge about the normal and abnormal

disease.

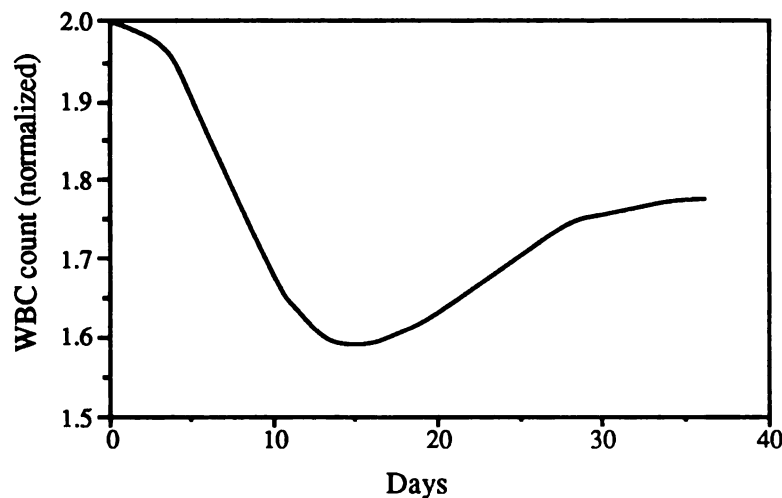


Figure 3.1 A patient's white blood cell count following a single dose of chemotherapy.

Patients receiving myelosuppressive chemotherapy drugs have a drop in their peripheral white blood cell (WBC) count over a period of 3 to 4 weeks. Depending on the particular drug, the WBC count may not return to the pretreatment value. An actual patient record is likely to have only two or three observations over this time period.

effects of chemotherapy on WBC counts. If we had this additional knowledge, we could then analyze the observations intelligently to determine the answers to these clinically relevant questions.

Changes in clinical signs and symptoms are caused by the intricate interplay between physiological and pathophysiological processes. Because processes exert their influence over time, observed changes give insights about these unseen, unmeasured, but pervasive underlying processes. Therapeutic maneuvers attempt to eliminate pathophysiological processes and to restore homeostatic processes. The current physiological state of the patient has direct and immediate therapeutic implications for two reasons:

1. Specific physiological abnormalities can suggest therapeutic actions that directly address the most serious problems.
2. The current physiological state provides prognostic information that can modify therapeutic goals, strategies, and options. For example, a leukemic patient with successful induction chemotherapy receives consolidation therapy in the hope that he may

achieve a complete cure; while another leukemic patient with unsuccessful induction chemotherapy faces salvage therapy in the hope that he may achieve an initial remission.

Because physiological processes are central to therapeutic decision making, the clinical abstractions that are most useful for therapy planning are based on physiological principles. Normal and abnormal physiology are the expressions of multiple, simultaneous, interacting processes. The ability to draw conclusions from time-ordered primary clinical observations requires extensive knowledge about the temporal relationships and constraints on the observations imposed by physiological principles. Without this explicit knowledge, the reasoning system can only describe the observed changes—it cannot infer the physiological implications of these temporal changes.

3.3 Summarizing Time-Ordered Clinical Data

Webster's dictionary defines the word *summary* as follows: "A short, restatement of the main points for easier remembering, for better understanding, or for showing the relation of the points; an abstract; an abridgement or compendium containing the sum or substance of a fuller statement" [47]. Three key aspects of a summary are explicitly highlighted in this definition:

1. A summary is *comprehensive*. It contains the essential features of the original facts.
2. A summary is *brief*. It is less detailed than the original facts.
3. A summary is *abstract*. It contains concepts represented by the data, rather than being a restatement of the data.

A summary is the result of two actions: (1) the selection of a relevant subset of features chosen from a larger list of observations or measurements, and (2) the replacement of specific features by more general statements that capture the essential concepts or implications of the original data. To perform these tasks correctly, we must use specialized knowledge. Because only some features are relevant, decisions must be made about which observations contain useful features and which ones can be ignored. Determining relevance requires knowledge about the underlying motivation for analyzing the patient's record. This

is a complex issue, because a patient's record may be analyzed for many reasons. For example, a physician making a therapeutic decision may choose to focus on the relationship between prior treatments and subsequent clinical outcomes, whereas the same physician considering a diagnostic decision for the same patient may concentrate on the evolution of new clinical features between treatments. The set of relevant features could be entirely different, even though the same patient record is analyzed. The presence of a specific unusual finding or implication may make initially irrelevant measurements important. For example, a sudden weight change may make previous routine weight measurements important. Thus, we cannot categorically predetermine which features can be discarded for a specific analysis. Defining the meaning of the *essential concepts* within the data is also problematic. What may be essential in one setting may be unnecessary in another setting. In cancer chemotherapy, understanding the dynamics of bone-marrow suppression is critical to effective therapy planning; in diabetes therapy, this concept is not useful to any therapeutic decision. The general statements that replace specific patient observations must be more than merely accurate; they also must be informative. Uninformative statements defeat the goal of a brief but comprehensive summary.

By summarizing a set of primary data, we lose information. The abstractions derived from primary data are insufficient to regenerate the initial data. For example, two arithmetic calculations, the mean and standard deviation, are two ways to summarize a set of measurements. To derive these *summary statistics*, we use the initial data. However, given only the mean and standard deviation, we cannot recreate the original data. Other features, such as the minimum and maximum observed values, also are lost. If these statistics capture an essential feature of the data required for subsequent analysis or interpretation, then they constitute an effective summary. Otherwise, important information has been irreversibly lost in the summarization process. An key implication is that there is no single *correct* summary that captures all essential features for all tasks. For some tasks, the mean and standard deviation are adequate; in other settings, they are not sufficient.

If summarization causes irreversible loss of information that may have been costly to obtain, then why summarize? In settings where data are sparse and expensive, it may not make sense to reduce further the available data by summarization. However, in most settings, the number of raw data is too great for the data to be manipulated directly. Only a small subset of data is informative for a specific problem. In many cases, key features are implied by relative relationships among multiple measurements. In clinical practice,

data frequently are redundant; multiple measurements confirm the presence of the same concept. An effective summary must draw a delicate balance between providing significant data reduction and eliminating important information. These two goals are inherently conflicting. The tradeoff is simplicity versus detail. The proper balance is both problem- and context-specific. For example, a patient with complex, multiple life-threatening medical problems is “seriously ill.” This gross summary statement is sufficient for family members, but is useless for consulting physicians.

Even for a rapidly changing system, the abstraction of temporal features need not be an essential feature of summarization. An atemporal description of the current system state could be critical for solving a particular problem. This type of description is called a *snapshot* summary. In Figure 1.3, a snapshot summary creates summary statements from a single column in the patient flowsheet. Although this snapshot might be useful in some settings, medical problem solving usually exploits key concepts embodied in the temporal evolution of a disease for diagnostic and therapeutic decisions (Chapter 2). A clinical summary that is devoid of temporal features is of limited usefulness as an aid to solving complex diagnostic and therapeutic problems. Medical decision-support systems require summarizations that capture important temporal abstractions.

In the remainder of this section, I shall examine previous research in summarizing time-ordered medical data. Each of the works used different techniques to detect and abstract temporal concepts in medical data. Two key concepts appear in the programs I shall present: (1) the need to understand the clinical contexts that exist when an observation is recorded, and (2) the need to represent expectations that change over time. Both concepts will figure prominently in the development of the methodology that I shall present in Chapter 5.

3.3.1 VM

The general features of Fagan’s ventilation management program, VM, were presented in Section 2.3.3.1. VM was a rule-based expert system designed to interpret on-line physiological data using IF-THEN rules to derive clinical abstractions from patient data. These rules encoded the knowledge about which findings had to be present for a clinical abstraction, such as hypertension, to be concluded. Context-specific rules were used to convert observations into context-free clinical abstractions. For example, a context-specific defini-

tion of hypertension was used to examine blood-pressure measurements; once hypertension was concluded, this clinical abstraction was no longer context-specific. This conversion of context-sensitive measurements into context-free clinical abstractions was a key contribution of VM. It permitted the rest of VM's reasoning, which used only the clinical abstractions to derive therapy-management conclusions, to be context-free even though the underlying problem domain was clearly context-sensitive.

One of VM's goals was to maintain a summary of the patient's physiological status over time. During each time interval, VM determined the set of clinical abstractions that were supported by the current patient data. VM then combined abstractions in adjacent intervals into one continuous interval. If there were two adjacent intervals during which VM concluded that the clinical abstraction *tachycardia* was present, VM replaced these two intervals with a single interval of tachycardia that extended over the duration of both smaller intervals. Because context-specific rules provided definitions that accounted for any context-specific differences, adjacent abstractions could be combined even if they occurred different clinical contexts.

VM provided a simple summary of a patient's clinical course using alphanumeric plots. Three types of summary displays were available: (1) a description of current conclusions, (2) a plot of recent conclusions over time, and (3) a plot of violated expected parameter measurements over time. Of the three displays, the plot of conclusions over time was most similar to a summary of the patient's medical record. This output listed clinical abstractions that had been concluded and combined by VM during the previous 6 hours of monitoring (Figure 3.2).

3.3.2 RX

The general features of RX were presented in Section 2.2.2.3. RX combined symbolic and statistical techniques to analyze sets of patient records for the presence of previously unknown causal relationships. RX derived abstracted interval events by examining features present in each patient visit. Each potential interval event type had a derivation tree, which was a hierarchically arranged set of features that had to be present for an instance of the interval event to be considered present in this patient. Conclusions lower in the derivation tree were called *support variables*, because their presence supported the presence of the interval event. Each interval event and its derivation-tree nodes contained

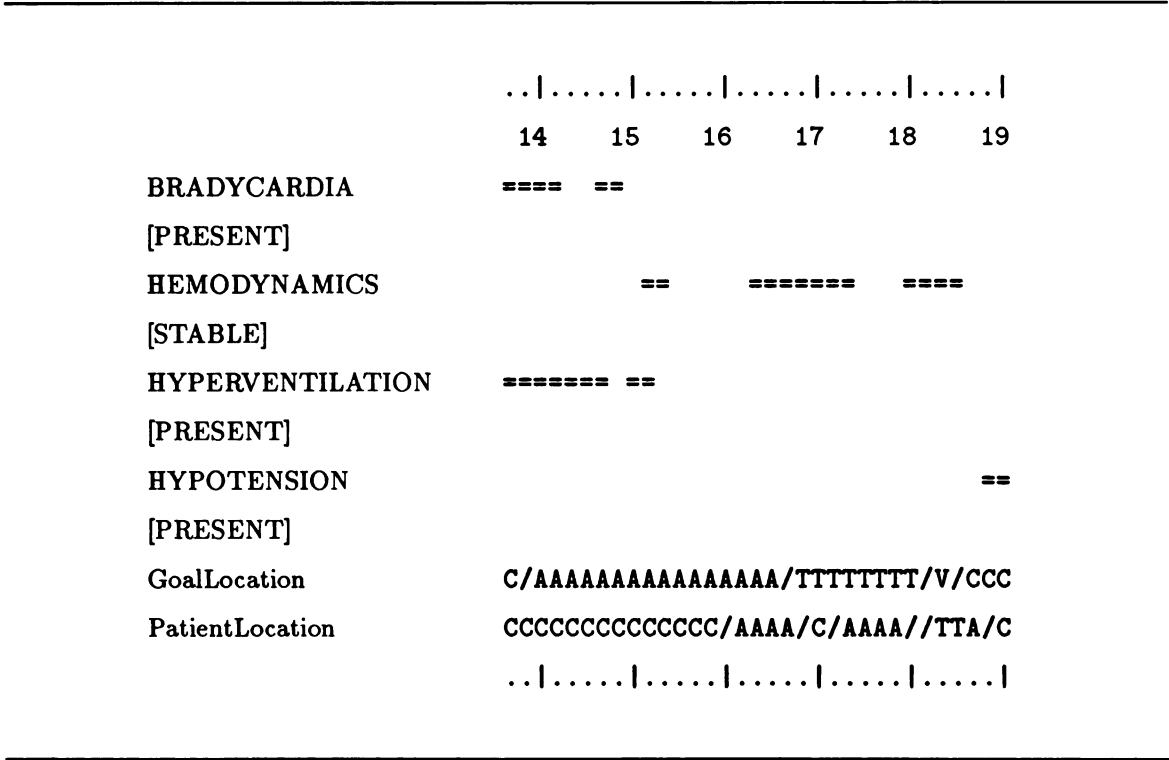


Figure 3.2 A summarization plot generated by VM.

VM listed conclusions made over the past 6 hours in alphabetic order. GoalLocation is VM’s conclusion of which state is considered ideal for this patient at this time. PatientLocation is the current patient state. Each mark represents 10 minutes during which the conclusion was present. (Adapted from [39, pg. 38].)

arbitrary LISP predicates connected by Boolean functions. Interval events were concluded using a depth-first traversal through the derivation tree. If all predicates in the derivation tree succeeded, an instance of that interval type was created.

A second key feature developed in RX was the use of observations from one point in the patient's record to support the conclusion of a clinical abstraction during another point in time. For example, the presence of urinary protein at a visit 7 days ago could be used to justify the clinical abstraction of nephrotic syndrome on the current visit, even if there was no value for urinary protein recorded for the current visit. Blum used this feature extensively to handle missing data. By using data from visits that were close to the current visit, RX derived more interval events than it could have derived using only the data from the current visit.

Blum developed a display program that plotted both patient findings and RX conclusions on an alphanumeric display (Figure 3.3) [14]. RX could display entities that were considered point events (such as a laboratory measurement) and abstractions that were considered interval events (such as an episode of hepatitis) on the same plot. With this program, point- and interval-based temporal relationships among various entities could be explored visually.

Many display programs generate simple plots of data stored in time-oriented databases. The RX program was unique in its ability to manipulate interval and point entities simultaneously. The display program had no temporal knowledge and performed no temporal inference. The interval entities plotted in Figure 3.3 were derived by RX. As with the plot generated by VM, the interpretation of the display is left to the physician. However, plots such as those in Figures 3.2 and 3.3 are summaries in that they both reduce data and increase comprehension of possible temporal relationships among the displayed variables.

3.3.3 Downs' Summarization Program

In both the VM and RX projects, summarization was not the main thrust of the research, although the investigators of both found the creation of simple summarization and visualization tools useful. Downs constructed a program the primary purpose of which was to summarize patient records automatically [37,38]. He used the same patient records that were used in RX, although the methodology he developed for his program was independent

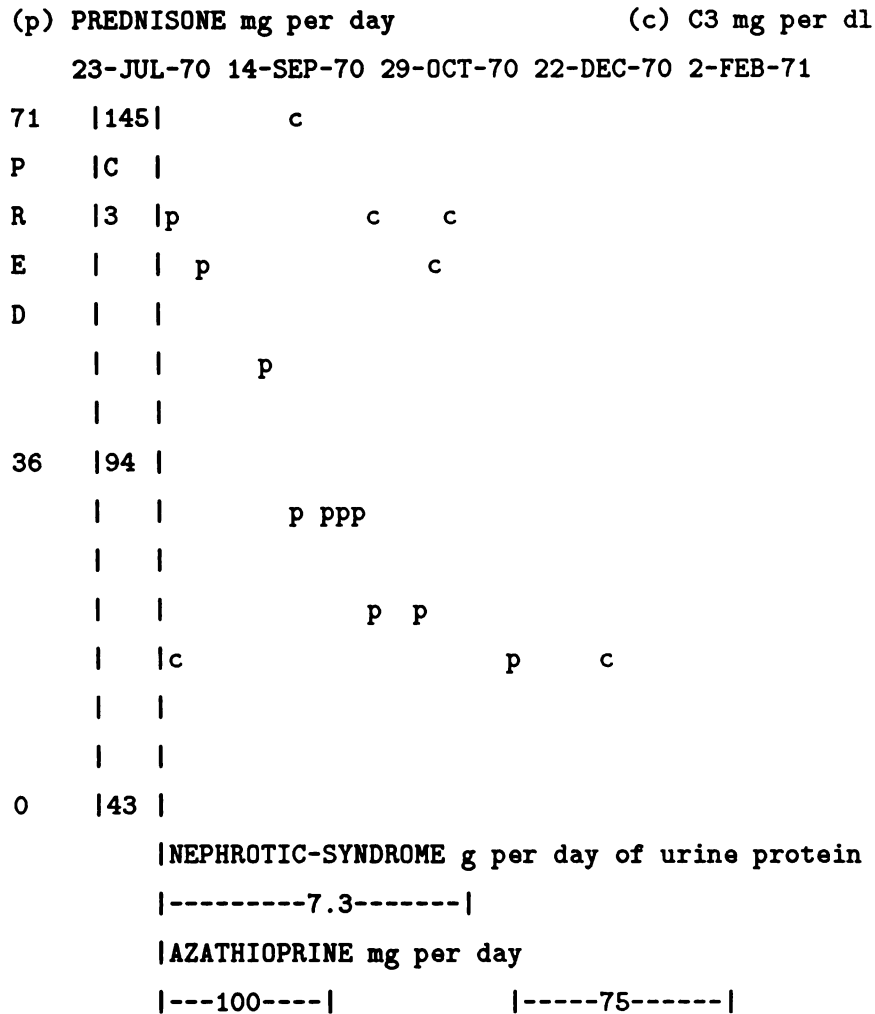


Figure 3.3 A sample patient record plotted by RX.

RX plots could contain both point events and interval events. In this plot, the lower-case letters represent point measurements of prednisone dosage (plotted using the letter *p*) and C₃ complement levels (plotted using the letter *c*). Lines represent interval events. In this example, the patient had one episode of nephrotic syndrome and two courses of azathioprine administration. (Adapted from [14, pg. 198].)

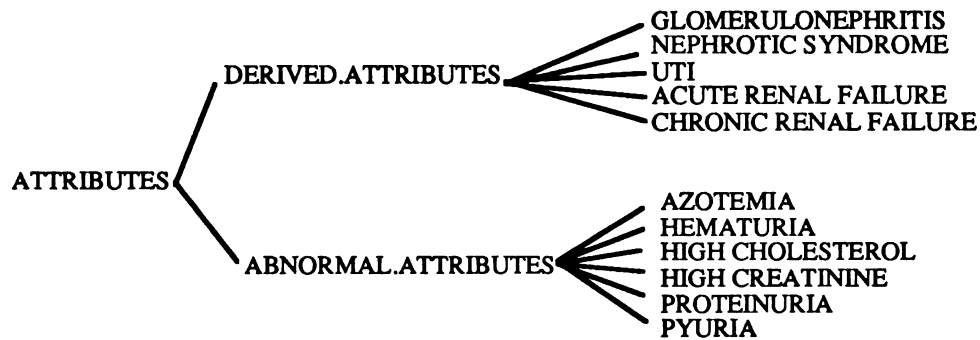


Figure 3.4 The knowledge-base structure in Downs' summarization program.

ABNORMAL.ATTRIBUTES were derived from patient measurements via a probabilistic scoring scheme. DERIVED.ATTRIBUTES were clinical states that were derived from ABNORMAL.ATTRIBUTES via the same scoring scheme.

of the RX research.

Downs divided the summarization task into two sequential phases: (1) the detection of abnormal patient findings, called ABNORMAL.ATTRIBUTES, and (2) the detection of potential disease states, called DERIVED.ATTRIBUTES (Figure 3.4). ABNORMAL.ATTRIBUTES represented clinically significant patient states that were concluded based on examination of raw patient data. Each ABNORMAL.ATTRIBUTE contained a list of potential DERIVED.ATTRIBUTES that could have caused the abnormal patient finding. When an ABNORMAL.ATTRIBUTE was detected, the list of DERIVED.ATTRIBUTES became the differential diagnosis to explain the occurrence of the abnormal findings. For each potential DERIVED.ATTRIBUTE on the current differential diagnosis, the system attempted to find sufficient evidence to confirm or deny the presence of the diagnosis. By using the data to trigger potential diagnoses and the diagnoses to direct the confirmation phase, Downs' system combined data-driven hypothesis generation and goal-driven hypothesis testing.

The key contribution of Downs' program was the combination of the odds-ratio formulation of Bayes' formula with the RX temporal predicates to determine whether sufficient evidence was found in the record to support the conclusion of ABNORMAL.ATTRIBUTES or DERIVED.ATTRIBUTES. Each concept had one or more predicates that represented evidence for or against the presence of the concept. If a rule succeeded, the system calcu-

lated a posterior odds ratio that updated the belief in the existence of the concept. If a predicate was neither proved or disproved, no update occurred. Downs extended the RX predicates to use evidence at other patient visits to confirm or deny the presence of an ABNORMAL.ATTRIBUTE or DERIVED.ATTRIBUTE at the current visit. If a likelihood threshold was exceeded, the system concluded that the finding existed at the current visit.

If Downs' summarization system concluded that an ABNORMAL.ATTRIBUTE or a DERIVED.ATTRIBUTE existed at the current visit, it assumed that that conclusion persisted until the next recorded visit. At the next visit, the program performed that same search for ABNORMAL.ATTRIBUTES and DERIVED.ATTRIBUTES. Unlike VM, Downs' program did not join two adjacent instances of the same conclusion into a single episode.

In addition to introducing Bayes' formula with temporal predicates as a method to confirm or deny the presence of temporal concepts, Downs implemented a novel user interface. Both VM and RX exhibited the limitations of the technology of the early 1980s. At that time, only 24- by 80- character alphanumeric displays were readily available. Plots, charts, and graphs were limited to the resolution of these simple output devices. Modern high-density, bit-mapped graphical displays, with a display density of 1000 by 1000 pixels, have removed these limitations.

Downs constructed a graphical interface to display ABNORMAL.ATTRIBUTES and DERIVED.ATTRIBUTES that were concluded by the summarization program (Figure 3.5). Shading and sizing represented the magnitude of the odds ratio and the estimated severity of the finding for each plotted attribute. Using a mouse pointing device, the user could select boxes on the display to obtain information about the temporal predicates and the odds-ratio values that were used to conclude the presence of the concept. In this way, Downs' program visually summarized the clinical abstractions that it had concluded.

3.3.4 IDEFIX

Downs' summarization program was succeeded by IDEFIX [30,31], which performed the same task on the same patient records but addressed some of the temporal deficiencies noted in the original program. The key problem addressed in IDEFIX was the assumption made in the earlier program that a concluded ABNORMAL.ATTRIBUTE or DERIVED.ATTRIBUTE persisted from the time of the current visit until the next pa-

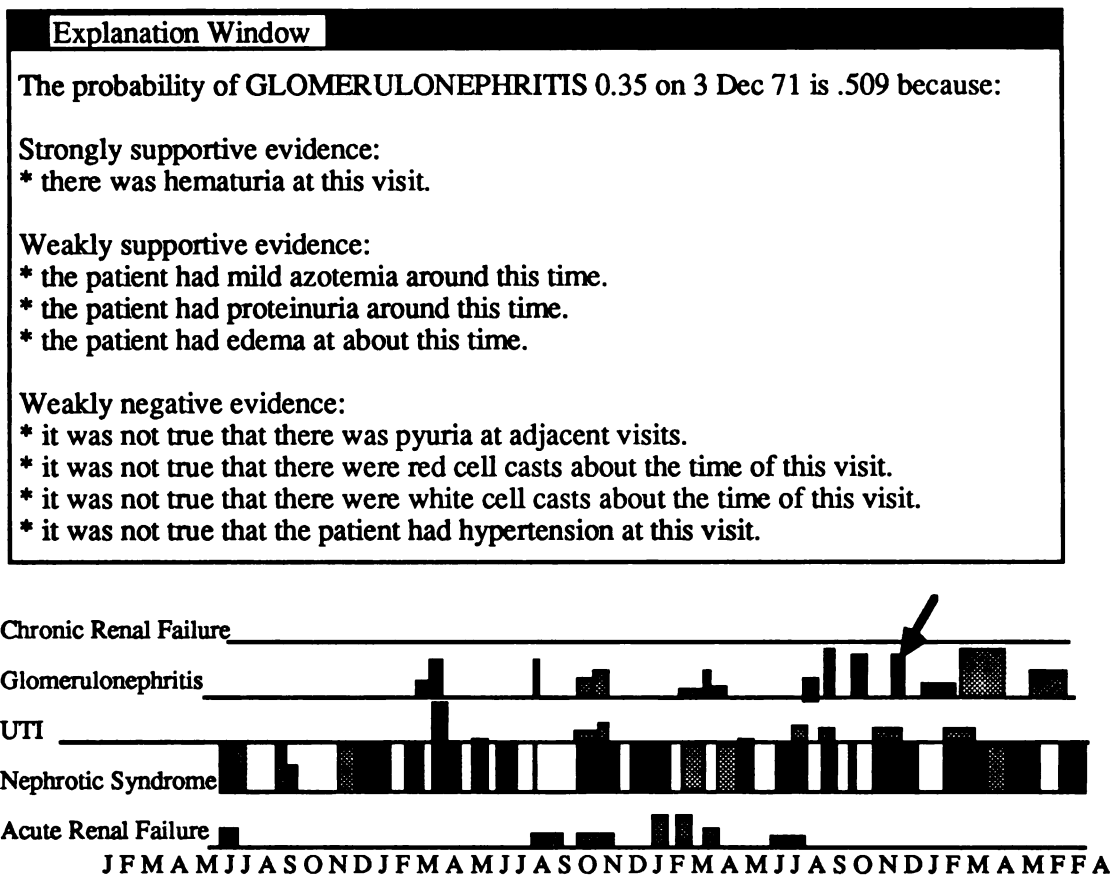


Figure 3.5 Graphical output of Downs' summarization program.

DERIVED.ATTRIBUTES concluded by the program were plotted as boxes against a time line. A box was displayed if the system concluded that the attribute was present at the time of that patient visit. The height of each box represented the severity of the finding; increased height denoted increased severity. The shading of the box represented the degree of belief in the conclusion (the value of the odds ratio); increased darkness denoted larger odds ratio. The box width corresponded to the time between the current visit and the next patient visit; increased width denoted longer intervisit interval. The explanation window appeared when the user selected a specific box on the plot. It displayed the calculated odds ratio and the predicates that matched or failed to match this finding. (Adapted from [37, pg. 6].)

tient visit at the same level of certainty and severity. A closely related problem was the assumption that a datum either contributed no weight or its entire weight for or against a conclusion. Both assumptions clearly are incorrect. During a long interval between visits, an abnormality may have resolved. Without additional observations, our level of certainty about the continued existence of the abnormality and its severity should decrease. The second assumption does not capture the notion that the usefulness of clinical data for a conclusion degrades slowly over time. Older data may provide evidence for a conclusion, but their contribution should be sensitive to length of time between when they were collected and the current visit. The loss of contribution is dependent on the dynamics of the data, the dynamics of the hypothesized conclusion, and the time between the data collection and the conclusion.

We have medical knowledge about the expected temporal duration of most illnesses and about the usual temporal validity of most clinical observations. IDEFIX extended the combination of the odds ratio and temporal predicates in Downs' program to capture this additional temporal medical knowledge. De Zegher-Geets introduced the notion of a *time-oriented probabilistic function* (TOPF) to represent uncertainty that changes over time (Figure 3.6). A TOPF encodes the change in the conditional probability for a conclusion over time. Unlike Downs' program, which assigned the full likelihood ratio to a conclusion when a temporal predicate succeeded, IDEFIX used TOPFs to modify the likelihood ratio. The use of time-sensitive probability functions provided a method to *down-weight* the strength of evidence over time. IDEFIX defined four standard TOPFs, linear-increasing, exponential-increasing, exponential-decreasing, and logarithmic. TOPFs were also used to represent the expected change in severity of a finding over time. For example, most episodes of urinary-tract infections resolve 3 to 7 days after treatment is started. Although a patient may have evidence of a severe urinary-tract infection at a visit, without evidence of a complication, the severity of that conclusion should diminish over the next week. Using TOPFs, IDEFIX could encode this expected drop in severity over time, whereas Downs' program assumed a constant severity until the next patient visit.

3.3.5 Lessons Learned

Different models, assumptions, and techniques were used in the systems presented in the previous sections. Their strengths and weaknesses played a major role in the devel-

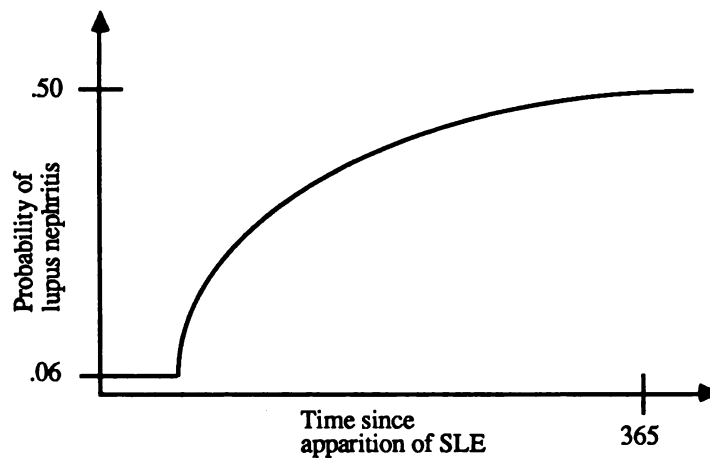


Figure 3.6 A time-oriented probability function in IDEFIX.

Time-oriented probability functions (TOPFs) enabled IDEFIX to change an observation's contribution to a conclusion as the interval between the observation and the conclusion changed. TOPFs also provided a method for IDEFIX to encode the expected change in the severity of a finding over time. The figure shows the TOPF associated with the predicate PRESENCE-OF-CAUSING-DISEASE SLE. In this example, the probability of lupus nephritis increases as the time since the onset of SLE increases. (Adapted from [31, pg. 42].)

opment of TOPAZ. VM introduced the notion of context-specific mapping of observations into symbolic abstractions. Abnormal ranges were changed as the clinical context changed. This feature was lacking in Downs' program and in IDEFIX. Suppose an otherwise healthy patient inadvertently received twice the usual dose of a myelosuppressive chemotherapy drug. The patient's WBC count surely would drop below the normal value. However, in this context, the low WBC count would not be unexpected. VM could potentially correctly represent the differences among a WBC count that is lower than the usual laboratory boundaries, a WBC count that is expected to be low, and a WBC count that is lower than expected. Because Downs' program and IDEFIX do not contain context-specific reasoning, both programs would conclude any value lower than the usual laboratory boundary to be abnormal. VM also introduced the notion of context-free reasoning once the context-sensitive mapping of observations to symbolic abstractions was performed. Context-sensitive rules changed as the clinical context changed, whereas context-free rules remained unchanged. Finally, VM combined adjacent intervals with the same temporal abstraction into a simple larger interval. As I shall show in Chapters 6 through 8, TOPAZ contains all these features.

Both VM and RX introduced the use of temporal predicates that used evidence at one point in time to support a conclusion at another point in time. Unlike VM, RX constructed abstractions using a hierarchy of supporting evidence. Lower-level abstractions, if present, would support more abstract concepts. With this hierarchy, RX could define complex clinical abstractions by decomposing their features into simpler abstractions. The hierarchical organization of abstractions was also used to partition the knowledge base in Downs' program and in IDEFIX. TOPAZ also organizes abstractions into a hierarchy and uses lower-level abstractions to derive more global clinical abstractions.

Both Downs' program and IDEFIX proposed unique models of temporal uncertainty. The combination of odds ratios with temporal predicates introduced by Downs' program and the addition of time-oriented probabilities functions in IDEFIX were new features not present in the older systems. The ability to represent the degree of belief in a conclusion that changes over time is a necessary feature of temporal data interpretation. TOPAZ also contains a method to capture its degree of belief in derived abstractions, although the model I have implemented in TOPAZ differs from the two models presented here.

3.4 Summarizing Cancer Chemotherapy

Most cancer chemotherapy consists of the administration of repeated cycles of multiple drugs. Other modalities, such as radiation therapy, frequently are given concurrently. The patient's therapeutic responses and toxic reactions to these maneuvers evolve over time. For example, 3 to 4 days after a patient receives a standard dose of mechlorethamine (Mustargen), a potent myelosuppressive chemotherapeutic drug, his WBC count will begin to drop. In most cases, the falling WBC count reaches a nadir 10 to 14 days later and returns to normal within 4 to 6 weeks (Figure 3.7, line A). Various deviations from this typical pattern have different clinical interpretations and therapeutic implications. A sharp initial drop in WBC counts with a low or normal nadir and a normal recovery implies an increased sensitivity to the toxic myelosuppressive side effect of the drug, with a normal bone marrow recovery after the drug has been eliminated (Figure 3.7, line B). A flat and extended nadir with a late recovery back to normal counts suggests prolonged myelosuppressive drug effect (Figure 3.7, line C). A WBC count that returns to a new steady-state level lower than the pretreatment level implies irreversible bone-marrow damage (Figure 3.7, line D). Cumulative toxicities may appear gradually across multiple courses of chemotherapy.

None of the patterns in Figure 3.7 are detectable through an examination of an isolated WBC value, yet each pattern has significant therapeutic relevance. In cancer chemotherapy, therapeutic options include reducing the dose of drugs, substituting one or more drugs, delaying treatment, or aborting further treatment. The risk of further untoward effects must be weighed against the risk of tumor growth. Cancer drugs frequently are given in doses up to the limits of the patient's tolerance. A decision-support system that is too quick to back away from treatment may deny a patient the only chance for a cure; a system that continues unmodified treatment in the face of serious problems may subject a patient to life-threatening complications.

In cancer chemotherapy, time clearly plays an important role in the abstraction and summarization of a patient's clinical course. The proper *interpretation* of a series of WBC measurements requires a detailed understanding of the expected temporal evolution of WBC counts following chemotherapy, the detection of a significant deviation from the expected pattern, and a method to derive the clinical implications of the observed deviations. The expected temporal evolution depends on many interacting factors, such as the type of cancer, the clinical stage of disease, the number of previous chemotherapy cycles,

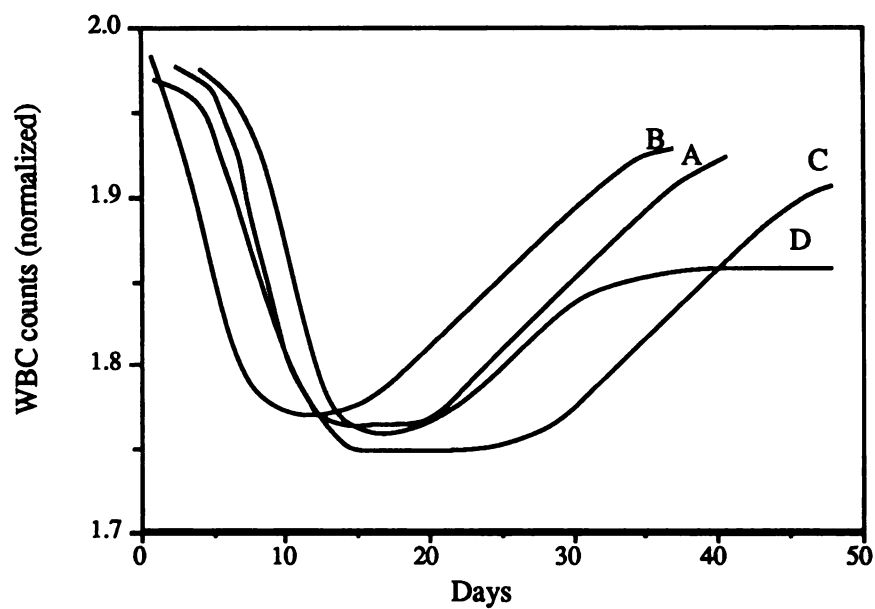


Figure 3.7 Mechlorethamine-induced changes in a patient's white blood cell count.

The typical response to a standard dose of mechlorethamine (nitrogen mustard) is depicted in line A. Various deviant responses that imply different clinical implications and result in different therapeutic decisions are shown in lines B, C, and D.

and the concurrent administration of adjuvant therapies. We must be careful to distinguish between two expectations: (1) what we expect to observe in the *average* or *typical* patient with the same clinical characteristics in this setting, and (2) what we expect to observe in *this* patient in this setting. Both expectations are useful. We can use the first expectation, called the *population expectation*, to detect instances where a patient seems to demonstrate difficulties. If we expect a typical patient to have moderate WBC-count depression following chemotherapy, then an observed low WBC count at that time in our patient is not worrisome. A profoundly reduced WBC count, however, would be an indication that this patient is not a typical patient. If we have prior evidence that this patient is particularly sensitive to chemotherapy, then the *patient-specific expectation* is that this patient will not tolerate another course of chemotherapy any better than he did the last course. In this situation, the dangerously low WBC count will match our patient-specific expectation. For this individual, the very low WBC count was expected; for the average patient, the same count would be unexpected.

3.4.1 OBOE

I developed a prototype summarization program, called OBOE, that predates the work I shall describe in Chapters 5 through 8 [57]. Like the later work, OBOE was designed to summarize patient records, but OBOE had a markedly different design. OBOE was an initial prototype; the problems I uncovered while creating OBOE led to the methodology I shall present in Chapter 5.

OBOE derived summary statements from a patient's computer-based medical record (Figure 3.8). Unlike the prose text output created by TOPAZ, the output from OBOE was a machine-readable data structure. These data structures were intended to be used by another program, called ONYX, that would use the abstractions created by OBOE to construct individualized chemotherapy plans for patients with very unusual clinical courses [71]. OBOE focused on finding recurring problems with the *administration timing* and *dosage* of chemotherapy drugs in patients who were being treated on a chemotherapy protocol.

OBOE used a two-pass design to create summary abstractions from records of patients receiving cancer chemotherapy (Figure 3.9). The first pass analyzed data from each visit for deviations from the *expected* clinical course. The second pass reviewed data

WBC X 1000		5.5	8.3	7.2	3.5	4.3	6.9	2.0	4.1
Platelets		281	117	420	288	163	349	396	
Chemo Name	VAM	POCC	POCC	VAM	POCC	POCC	VAM	POCC	POCC
VP-16	175			140			140		
Adria	115			85			85		
MTX	70			60			60		
CTX		1300	1300		1300	1300			1300
CCNU		140			100				80
VCR		2	2		2	2			2
PCB		225	225		200	200			
Day	15	6	13	3	14	31	21	12	19
Month	Jun	Jul	Jul	Aug	Aug	Aug	Sep	Oct	Oct
Year	84	84	84	84	84	84	84	84	84

[PROTOCOL.MODEL

[MODEL (STATE.OCCUR ((“Protocol Not Completed” ALWAYS]

(SUBMODELS

((CHEMOTHERAPY.MODEL

[MODEL [STATE.CHANGE ((GOT.WORSE (84 8 3]

(STATE.OCCURRED ((“Abnormal Chemotherapy” (6 “out of” 8 “intervals”))
 (“Normal Chemotherapy” (2 “out of” 8 “intervals”]

(SUBMODELS

([POCC.MODEL

[MODEL [STATE.CHANGE ((GOT.WORSE (84 8 24]

(STATE.OCCURRED ((“Abnormal POCC cycle” (3 “out of” 4 “intervals”))
 (“Normal POCC cycle” (1 “out of” 4 “intervals”]

Figure 3.8 A sample patient record and OBOE output.

A OBOE summary was a nested data structure composed of models and submodels representing successive levels of abstraction. Submodels contained abstractions derived from shorter temporal intervals. The OBOE output was a machine-readable data structure, rather than prose designed to be read by a physician.

from multiple visits for abnormal temporal patterns.

The first pass encoded the *expected* clinical course as an augmented transition network (ATN) (Figure 3.10). Each node in the ATN represented a potential state corresponding to stages in the care of patients on a particular protocol. Included in the ATN were nodes for states that represented expected protocol variations. The ATN was used to perform the conversion of raw patient data into symbolic values using context-specific interpretation routines associated with each ATN node. For example, for one clinical context, an ATN node might convert a WBC-count value of 11.0 to “normal”; in another context, a different ATN node might convert the same value to “elevated.”

The second pass used standard IF-THEN rules to perform the aggregation and analysis of the symbolic conclusions from the ATN. The rules analyzed aggregates of conclusions to detect trends that occurred over longer periods of time. Cancer-chemotherapy protocols provide a natural hierarchy of time intervals for abstraction: individual visits, chemotherapy subcycles, chemotherapy cycles, and the entire protocol. OBOE used this temporal hierarchy to construct interpretations of ATN conclusions based on successively longer temporal intervals. Rules attempted to match temporal patterns within a set of ATN conclusions at each level in the temporal hierarchy. The rule conditions examined a set of conclusions for features (such as increasing or decreasing trends) or counted the number of unusual events during an interval. Starting at the lowest level in the temporal hierarchy, the rule set examined the set of visits within all subcycles, then the set of subcycles within all cycles. A final pass examined the set of visits, subcycles, and cycles within the protocol. In this way, larger intervals were analyzed by the rules.

OBOE revealed a number of deficiencies, which I shall discuss in greater detail in Chapter 9. The key problem was the ATN-based representation of the *expected* clinical course. Each potential patient state during chemotherapy required an explicit ATN node. Thus, for even simple protocols, this approach required hundreds of nodes. Because each node could represent only one potential combination of interacting events, complex interactions resulted in a rapid proliferation of new patient states. If a patient state was encountered that had not been explicitly encoded, the ATN had no node to represent that situation. In that case, OBOE assumed that the patient was taken off the protocol, and further processing was terminated. This model of the expected clinical course and expected patient findings was not sufficiently robust to express detailed underlying knowledge about changing expectations over time. In the methodology I shall present in Chapter 5, a more

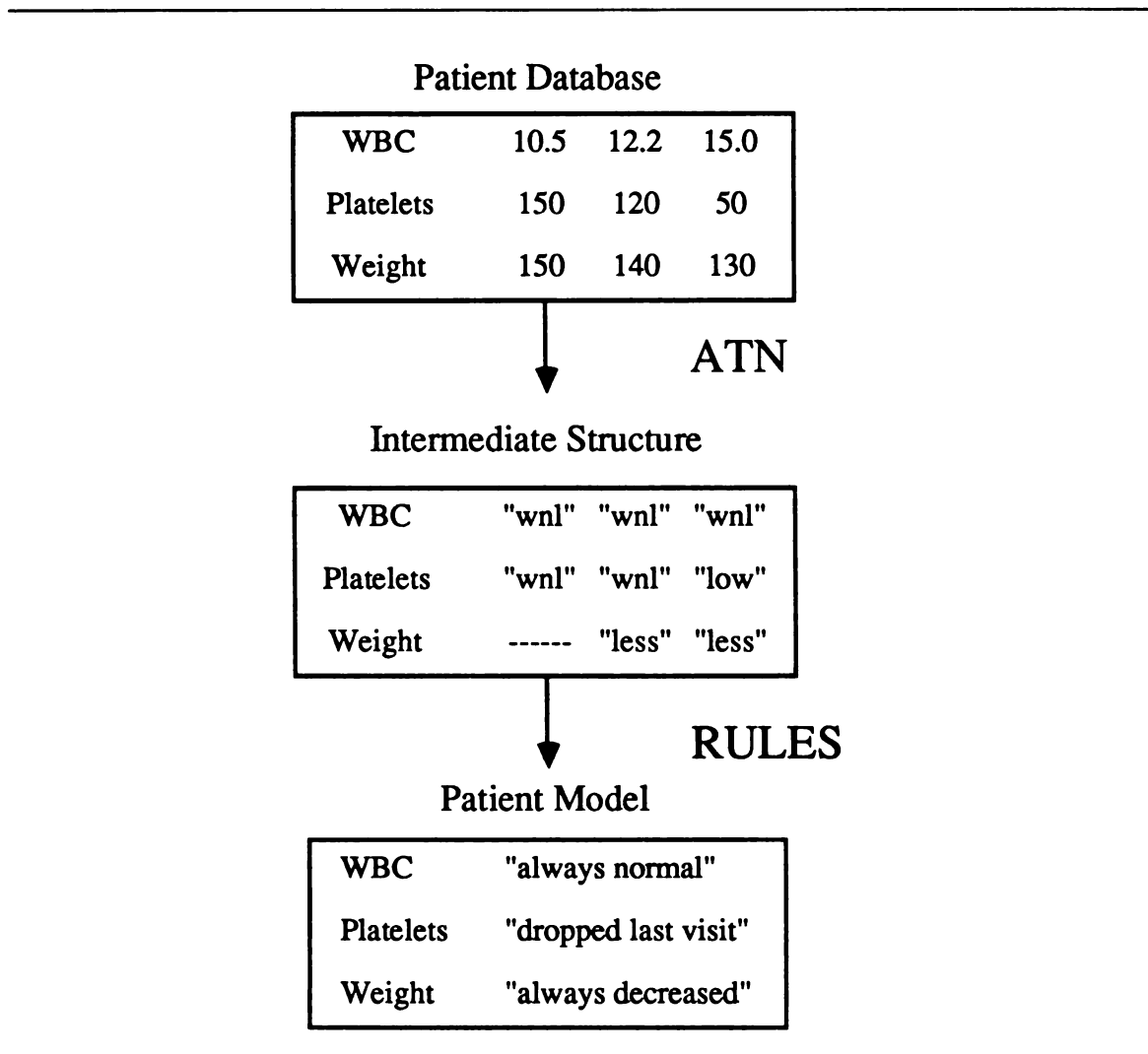


Figure 3.9 Two-pass system design in OBOE.

The two-pass system in OBOE communicated via an intermediate structure. In the first pass, an augmented transition network (ATN) performed the numeric-to-symbolic interpretation; in the second pass, a rule base performed temporal pattern matching.

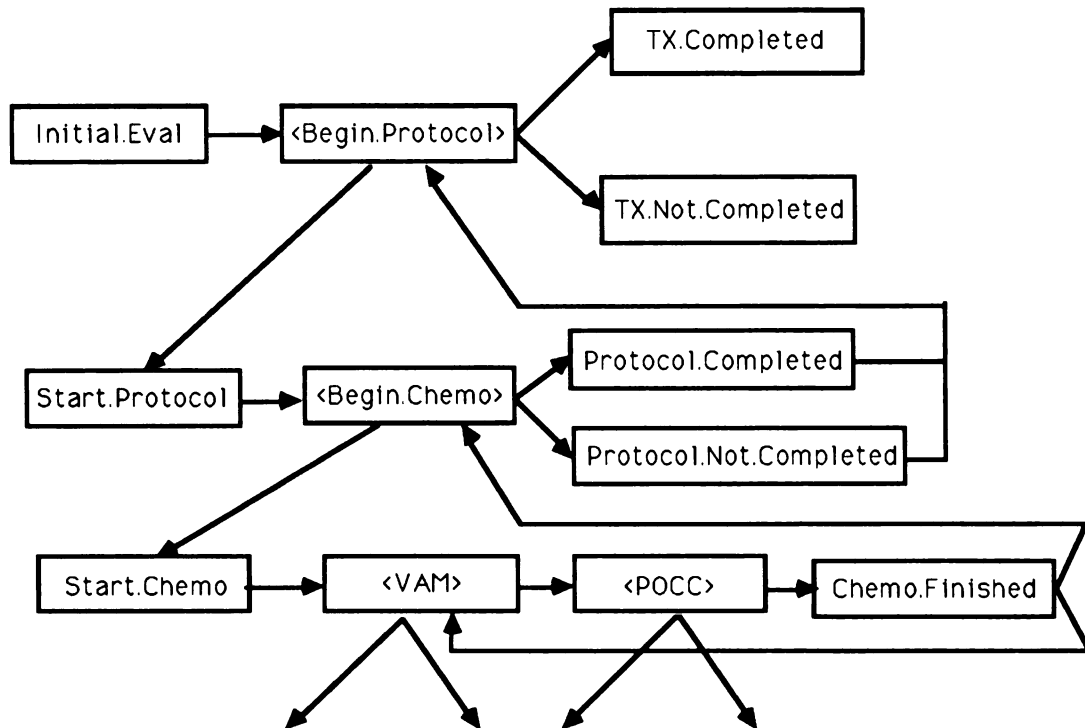


Figure 3.10 OBOE augmented transition network encoding of treatment states.

A fragment of the augmented transition network (ATN) in OBOE that encodes the initial states in a typical chemotherapy protocol. A patient is initially evaluated (INITIAL.EVAL), then begins a protocol (<BEGIN.PROTOCOL>). Later in his clinical course, the patient will either complete treatment (TX.COMPLETED) or end treatment without finishing all the expected courses of therapy (TX.NOT.COMPLETED). The ATN node <BEGIN.PROTOCOL> calls a SUBATN that encodes details about the specific chemotherapies in a particular protocol.

robust model of the expected clinical course than could be encoded by an ATN is central.

3.4.2 Implications for Summarizing Time-Ordered Medical Data

A summary must be a delicate balance of competing and contradictory demands. Informative and concise summary statements are both problem- and context-specific. In medical problem solving, temporal concepts contain essential diagnostic and therapeutic information that cannot be ignored in the construction of a summary of a patient's medical record. Temporal-reasoning and temporal-representation issues must be addressed in any computer-based decision-support program that reasons from the medical record.

Multiple summarization methods currently exist. Each method contains particular assumptions and has specific requirements. The characteristics of the summarization or the features of the underlying system may dictate which methodology is most useful. No single approach will be capable of expressing all the relationships and knowledge that may be needed in all circumstances. A method that integrates multiple methods permits the exploitation of the best features of each approach. This notion is the central theme of the methodology I shall propose in Chapter 5.

CHAPTER 4

The ONCOCIN Temporal Network

Each patient manifests signs and symptoms of his disease and of his responses to therapy with a unique intensity and tempo. Because a patient's course never exactly matches the classic description of the disease or of the ideal response to treatment, clinicians routinely deal with the wide variability of disease manifestations. To capture the complexity of a patient's illness, clinicians identify key periods of time in which important clinical events occur. Examples of key clinical events are recent clinic visits, previous treatment courses, and past hospital admissions. With these abstract temporal aggregates, the physician organizes and thinks about the patient's history in terms of events that are more meaningful than are simple dates or times. Experiments in cognitive psychology have shown that people structure and organize a wide variety of temporal relationships [43,35]. Although no unified model exists, psychologists note that *events* and *intervals* are powerful organizing abstractions that provide temporal ordering for a succession of related events [95]. Most of the temporal models that I presented in Chapter 2 were constructed from some notion of points and events. These two temporal abstractions, although not sufficient to capture the whole range of temporal complexity, impose significant structure on a dynamic world.

Developers of early medical database systems recognized the need to associate

time information with data that may be obtained repeatedly. Wiederhold first described the Time-Oriented Databank (TOD) model, which was subsequently implemented for a national rheumatology database [129,44]. An example of the standard TOD structure was shown in Figure 1.3. Each value in the database has an associated time-stamp that distinguishes multiple determinations of the same attribute. Blank cells represent times when not all attributes were recorded. Because each value has a time stamp denoting when that value was observed, databases designed using the TOD model are historical databases (Section 2.2.1.1.3).

Although a TOD-based database encodes a time-stamp with each observation, a close examination of a typical TOD record reveals significant missing temporal information. With this simple representation of temporal information, there is no method to appreciate meaningful groupings of data, such as sequential clinic visits, clinical exacerbations, or other clinical events that may relate multiple columns of data. The basic TOD structure does not contain the additional information needed to partition data into pertinent time intervals that represent the relevant abstract clinical temporal entities used by physicians. For example, “all WBC and platelet values that were obtained during the last two episodes of chemotherapy” is a useful collection of data based on meaningful clinical events.

In this chapter, I shall describe the difficulties with temporal issues that we encountered in the first version of ONCOCIN, an expert system that provides therapy advice to physicians caring for patients who are receiving experimental cancer treatments. Important clinical events—such as repeated chemotherapy cycles and concurrent radiation therapy treatments—were difficult to represent in the first version of the program. This difficulty resulted in the need for awkward and specialized methods to perform any reasoning that depended on temporal relationships between key clinical events. When ONCOCIN was reimplemented to run on single-user graphic workstations, we had the opportunity to rectify some of the temporal-reasoning problems that we had identified in the initial program. I shall describe a new database structure, called the temporal network (TNET), that I designed and implemented to address the recurring sources of difficulties with ONCOCIN’s temporal-information processing. TNET associates intervals of time and collections of data with clinical events. With this structure and a temporal query language for data retrieval, called TQUERY, ONCOCIN can access the patient’s database using contextually dependent queries based on the presence of key clinical events or temporal constraints. In the examples that follow, I use temporal entities that are meaningful in cancer chemotherapy

(such as VISIT, CHEMOTHERAPY, RADIATION-TX), but the names of the entities in a temporal network may be defined for each application domain. For example, TNET was used, without modification, to represent temporal entities in a therapy-advising expert system in hypertension clinical trials [91]. In Chapter 7, I shall describe an extended version of TNET, called ETNET, that supports additional temporal capabilities required by TOPAZ but not needed in ONCOCIN.

4.1 ONCOCIN

ONCOCIN is a decision-support expert system that aids physicians in the clinical management of patients who are receiving complex experimental cancer treatments under the guidance of a research protocol [11,123]. A protocol contains carefully written instructions to the physician detailing how a patient is to receive a new cancer treatment. Protocols are designed to produce a set of patients with the same disease who have been treated almost identically. A typical research protocol contains two or three possible treatments; a patient is assigned to one of these treatments at the first visit to the physician. The patient outcomes from one treatment are compared statistically to the patient outcomes for the other treatments. Due to the nature of cancer treatment, patients receive multiple drugs in various combinations in alternating sequences or cycles over many weeks or months while also receiving other treatment modalities, such as radiation therapy. These protocols contain extremely complex therapeutic guidelines that require continuous judgmental reappraisal of the patient's disease and of his response to treatment.

In this chapter, I shall discuss two prototype versions of ONCOCIN. Both prototypes perform the tasks described in the previous paragraph, but they differ substantially in internal and external design. The first ONCOCIN prototype, which I shall call mainframe-ONCOCIN, was developed between 1979 and 1983 on a large mainframe computer. In 1983, ONCOCIN was transferred to single-user graphical workstations. In response to deficiencies we discovered while developing mainframe-ONCOCIN [93], ONCOCIN was redesigned. I shall call the second prototype workstation-ONCOCIN (the name ONCOCIN refers to both versions). Although the names used to distinguish the two prototypes emphasizes the hardware distinctions, the major changes were made in the underlying program design, data structure, and interface.

Using a rule-based approach, ONCOCIN has been implemented and used success-

fully in the outpatient oncology clinic at Stanford University since 1981. A performance evaluation showed that mainframe-ONCOCIN provided therapeutic advice at a level of expertise similar to that of Stanford oncology fellows [50].

Administering cancer chemotherapy involves giving sets of drugs. A set of drugs that are given together is called a *chemotherapy regimen*. More than one regimen can alternate or intersperse to form complex sequential patterns. In total, a patient may receive as many as 7 to 10 different drugs along with nondrug treatments, such as radiotherapy. A therapeutic plan determines which drugs, if any, are to be given at a particular visit to the clinic. If drugs are to be given, the plan must specify the dosage for each agent. Subsequent reasoning may reveal that, although drugs are due to be given at this time, clinical factors prevent their safe administration. When this happens, ONCOCIN must determine whether the intended treatment can be delayed until another visit, whether an alternative treatment is permitted by the protocol, or whether the treatment can be skipped and care continued as though treatment had been given. Each of these choices have ramifications for subsequent reasoning on follow-up visits. An important source of decision-making information is the patient's response to previous administration of drugs. Because chemotherapy regimens tend to involve recurring cycles, ONCOCIN consults the past patient record for information associated with prior drug administrations. ONCOCIN contains many IF-THEN rules that embody an understanding of the temporal relationships among clinical events occurring during the course of treatment.

Figure 4.1 shows a typical rule, found in both ONCOCIN prototypes, that specifies complex temporal relationships among features in the patient record. Note the relative timings expressed in the rule premises. For this rule to succeed, the patient must be at the *start* of a new chemotherapy cycle that *immediately follows* an aborted cycle. In addition, there has to have been more than one previous chemotherapy cycle at sometime in the past. Finally, the WBC count has to have increased from the last chemotherapy cycle. A simple TOD patient record would require a series of expensive sequential searches to determine the dates and values needed to evaluate this rule. Because there is no additional structure, most of mainframe-ONCOCIN's temporal queries require time-consuming serial searching of the patient database.

Despite mainframe-ONCOCIN's ability to perform at an expert level, the system's representation of temporal relationships was inefficient, inflexible, and domain-specific. Specialized structures and hand-crafted routines had to be built to handle each temporal task.

-
- If 1) This is the start of the first cycle after an aborted cycle, and
- 2) The current WBC count (in thousands) is between 2 and 3, and
 - 3) The total number of cycles of chemotherapy given (including the present cycle) is not 1, and
 - 4) The patient's WBC count has increased since the last start of chemotherapy,

Then Conclude that the dose attenuation due to low WBC counts is 75 per cent.

Figure 4.1 Sample translated rule from ONCOCIN.

This is an English translation of an ONCOCIN rule that appears in both prototypes. The actual LISP code that implements this rule is shown in Figure 4.2(a) and (b). Note that the rule refers not to specific calendar dates but rather to temporal relations among important clinical events, such as the start of a cycle following a previous aborted cycle.

Figure 4.2 shows the LISP code fragment that implemented the rule in Figure 4.1. Although the predicates from mainframe-ONCOCIN were more compact, much complexity was hidden within highly specialized routines. Routines such as NTHAFTER were specially tailored to solve specific temporal problems. Each new temporal task required a new routine that handled the new situation. The functions did not provide users or programmers with insight into the nature of the temporal processing. In workstation-ONCOCIN, TNET was created to support the explicit representation of clinical events and simultaneously to facilitate effective patient-data retrieval.

A related mainframe-ONCOCIN problem involved the relationship between data-storage requirements and data-retrieval performance. TOD databases require all attribute values to be associated with a time stamp that denotes when the attribute value was valid. Attribute values that do not change from one time to the next time usually are handled in one of two ways: (1) either the unchanged value is reasserted each time, or (2) only values that change from the most recent measurement are asserted. The first method results in data redundancy and large storage requirements if most attributes do not change values. The second method requires special searching procedures if there is a request for an attribute value on a date for which no value was recorded. In this setting, the system must

<pre> IF (AND(NTHAFTER ABORT 1) (LBETWEEN WBC 2 3) (\$ISNT CYCELCOUNT 1) (\$IS WBCINCREASED)) THEN (CONCLUDEVALUE WBCATTEN 75) </pre> (a)	<pre> IF (AND(EXISTS (CHEMOTHERAPY PREVIOUS) (SCOMPARE IS (VALUE.OF VISIT.TYPE \$) 'ABORT)) (NCOMPARE LBETWEEN WBC.VAL 2 3) (NCOMPARE > CYCLE.NUMBER 1) (SCOMPARE IS WBCINCREASE 'YES)) THEN (WBCATTN 75) </pre> (b)
---	---

Figure 4.2 LISP predicates in the two ONCOCIN prototypes.

These are two versions of ONCOCIN rules that implement the rule in Figure 4.1. The LISP code in (a) is from mainframe-ONCOCIN; the LISP code in (b) is from workstation-ONCOCIN. Only workstation-ONCOCIN contains the temporal network (TNET). The mainframe-ONCOCIN rule is more compact, because hand-crafted routines hide the temporal complexity of the underlying comparisons. Although the workstation-ONCOCIN rule is more verbose, generalized routines—such as EXISTS, SCOMPARE, and NCOMPARE—are used. SCOMPARE performs symbolic comparisons; NCOMPARE performs numeric comparisons. Use of the TNET query language, TQUERY, is demonstrated in the first predicate in the workstation-ONCOCIN rule. TQUERY is presented in Section 4.4.

search backward in time until the last recorded attribute value is found—the system must perform expensive serial searching of the patient database. Because mainframe-ONCOCIN frequently had to perform such searching, a solution to this problem was required. In addition to allowing effective data retrieval, TNET supports efficient patient-data storage.

4.2 The ONCOCIN Temporal Model

ONCOCIN provides therapy advice to physicians caring for ambulatory patients receiving outpatient cancer chemotherapy. In this setting, a patient receives care during prescheduled clinic visits. The physician makes a therapeutic decision after he interviews and examines the patient, reviews new and past laboratory data, and consults with other health-care staff responsible for concurrent treatment modalities or specialized services. When he makes a treatment decision, therapy is initiated and a follow-up visit is scheduled. If unexpected complications occur, the patient is seen before the prescheduled follow-up visit. The frequency of planned visits depends on the particular chemotherapy, but generally a patient is seen on a regular schedule (once per week or once per month).

ONCOCIN's temporal model assumes that decision making occurs only during outpatient clinic visits. ONCOCIN models the clinic visit as the atomic temporal event for therapy planning. Even if the recommended drug is to be infused intravenously over several hours or is to be administered orally over several days, ONCOCIN models the decision-making task as determining which drugs are to be initiated at this clinic visit. Other events, such as the actual infusion time for a drug, are not modeled because they have no effect on the decisions that the physician must make. Thus, although there are temporal events with durations shorter than that of a clinic visit (modeled as 1 day), these are not modeled or represented in ONCOCIN because they do not contribute to the domain task.

A patient generally alternates between two or more chemotherapy regimens. Clinic visits are associated with the chemotherapy regimen that was administered, or was *in effect*, when the patient was seen. This grouping of visits according to chemotherapy regimen is particularly useful in therapy planning that bases a current recommendation on the outcome of a previous administration of the same chemotherapy. This treatment-specific review of past patient data is done frequently for patients receiving cancer chemotherapy, so ONCOCIN frequently requests only those data gathered during episodes of similar therapy. In treatment regimens that alternate between two or more chemotherapies, the

previous administration of the same chemotherapy will not have occurred at the previous clinic visit. There may have been many intervening visits for the other chemotherapy or for unscheduled examinations. Therefore, although ONCOCIN can identify the characteristics of the data it requires—such as the last time the patient received a particular regimen—it cannot be certain in advance which visits contain the relevant data.

I created TNET to assist ONCOCIN in the retrieval of context-specific patient data. ONCOCIN makes complex data comparisons in order to determine the patient's clinical course during a previous episode of chemotherapy. Because ONCOCIN uses rules for making inferences, these context-sensitive temporal comparisons appear as rule predicates. I designed the TQUERY language to permit expression of context-sensitive data comparisons efficiently and intuitively. TQUERY provides the language in which data are stored and retrieved from TNET. The ONCOCIN rule predicates strongly determined the TNET objects and the form of the temporal query language. For example, TNET can represent events with a duration of less than 1 day. However, because ONCOCIN never reasons about events of smaller duration, TNET never creates these objects. As a model of reality, this assumption is clearly incorrect. For the ONCOCIN domain task, this limited temporal model is sufficient. TNET and TQUERY embody these assumptions.

4.3 The Temporal Network

ONCOCIN models the critical events that occurred during the patient's past history. The purpose of TNET is to allow workstation-ONCOCIN to retrieve relevant information about a patient's past. For example, to understand a patient's response to past treatment, ONCOCIN may need answers to these representative questions:

- What was the last [first] recorded WBC count?
- What was the first [last, highest, lowest] platelet count during the last three cycles of chemotherapy?
- What was the lowest platelet count ever recorded during any administration of VAM¹?
- Was there any hemoglobin value less than 8.0 mg/dl during the previous [any, current] course of radiation therapy?

¹VAM is a chemotherapy regimen consisting of three drugs: vincristine, adriamycin, and methotrexate.

TNET supports temporally dependent queries such as these. Note that these queries refer not to specific dates, but rather to relative temporal order. As the patient's clinical course proceeds, the *last* WBC count will be one obtained on a different absolute calendar dates. To formulate a query that requires the last WBC count, the user should not need to know when the value was obtained, but rather should need to refer to only the logical temporal position of the desired data. Figure 4.2(b) shows the LISP code for workstation-ONCOCIN that incorporates the temporal network system. This LISP code implements the rule in Figure 4.1. In the workstation-ONCOCIN version of the rule, no specialized functions are used. This use of more generic functions results in more verbose rule predicates, because specialized functions submerge the complexity of the task into the LISP code that implements the function. With the new rule language developed for workstation-ONCOCIN, the nature of the temporal task is directly discernible (see Section 4.4 for a description of the query language syntax).

What features does TNET make possible that are not present in a standard TOD-based patient record? When data have been stored with only date information, both TNET and TOD can retrieve information for a specific time or date key. The TOD system, however, cannot retrieve information based on aggregates of time. TNET provides a structure to represent multiple views of intervals of time, so that the patient database can be queried using temporal entities more meaningful than are simple times and dates. The temporal network permits temporal queries based on temporal abstractions, such as visits or cycles. These domain- and problem-specific abstractions are not represented in the basic TOD database.

4.3.1 Logical Implementation

TNET is an extension of the patient database and is one of three subsystems that constitute workstation-ONCOCIN (Figure 4.3). Two-way communication for database assertions and queries is permitted between TNET and the inferencing subsystem, the ONCOCIN REASONER. All other modules, such as the ONCOCIN INTERVIEWER, have only one-way querying privileges. An ONCOCIN module retrieves information by querying TNET. A valid TNET query must contain the name of the desired attribute and the temporal-context criteria that must be satisfied for the data to be applicable (Section 4.4). TNET decodes the temporal specifications from the REASONER and retrieves the re-

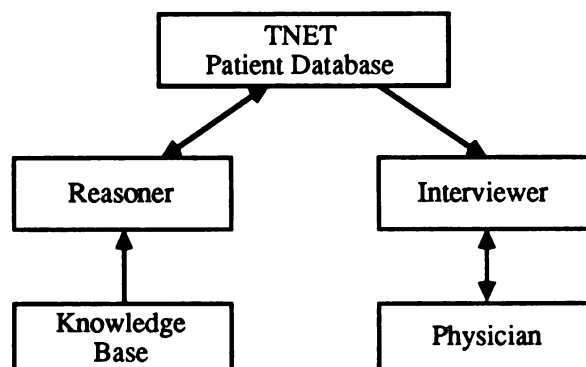


Figure 4.3 The general design architecture of ONCOCIN.

The temporal network (TNET) is an independent process that supports temporal queries from the ONCOCIN REASONER, the expert-system portion of ONCOCIN, and from the INTERVIEWER, the user-interface subsystem. Only the ONCOCIN REASONER can assert new data into TNET.

requested items from the patient database. The REASONER asserts new data items into the temporal network. The temporal network is then responsible for correctly placing the new data into the patient record.

The temporal network is a structured collection of nodes, called TNODEs. Each TNODE represents an interval of time during which an important clinical event was happening to the patient. For example, a TNODE in Figure 4.4 represents a period of time during which the patient received VAM chemotherapy. Another TNODE represents a later period of time when the patient received POCC² chemotherapy. TNODEs are distinguished by a type label and an optional subtype label. VAM and POCC TNODEs have the same type (CHEMOTHERAPY) but different subtypes (VAM or POCC). The labels used for types and subtypes are domain-specific. TQUERY selects relevant data subsets based on the occurrence of TNODEs that have the desired type or subtype labels. For the most effective use of TNET, the choice of type and subtype labels should reflect temporal concepts and intervals that segregate data into useful subsets or partitions.

A TNODE has two exclusive states: (1) *open*, which represents an interval that is currently happening, or (2) *closed*, which represents an interval that has concluded. Each patient has a unique TNET that contains numerous open and closed TNODEs that

²POCC is a chemotherapy regimen comprising four drugs: procarbazine, oncovin, cytoxan, and CCNU.

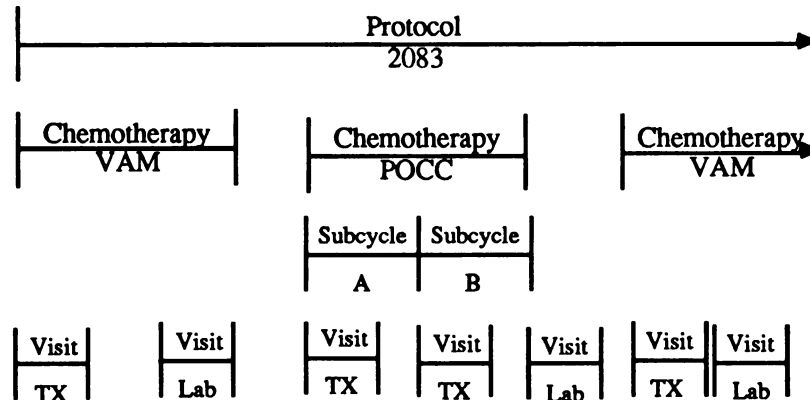


Figure 4.4 A TNET showing alternating courses of chemotherapy.

This patient received VAM chemotherapy alternating with POCC chemotherapy. In this figure, type labels appear above the line and subtype labels appear below the line. POCC cycles also contained A and B subcycles. The patient also had two types of visits, treatment visits (denoted by TX subtype) during which chemotherapy was given, and laboratory visits (denoted by LAB subtype) during which only laboratory tests were performed. In this figure, time advances from left to right.

represent the key events in that patient's past and present clinical history. In Figure 4.4, closed TNODEs have a vertical bar as their right boundary, whereas open TNODEs have a right arrow for a right boundary. Events of the same type that occur sequentially are linked and are called sib-nodes. In Figure 4.4, the three chemotherapy TNODEs are sib-nodes representing sequential episodes of chemotherapy cycles. Likewise, all of the visit TNODEs are sib-nodes, represent sequential clinic visits.

Patient data are stored in temporal-network nodes. In the oncology clinic, patients have data collected during clinic visits, so most database information is stored in VISIT TNODEs. Some patient attributes are stored in TNODEs that span multiple visits. For example, the value of the attribute CYCLE-NUMBER is stored in a CYCLE TNODE rather than being repeated in each VISIT TNODE within the temporal duration of the CYCLE TNODE. Any query to a VISIT TNODE for the value of the CYCLE-NUMBER attribute will return the value for this attribute stored in the CYCLE TNODE that temporally includes the queried VISIT TNODE. This storage technique allow attributes with meaningful durations longer than a single visit to be stored once in the proper interval and to be accessible to all children TNODEs that are temporally contained duration of the

parent TNODE. Static data are stored in the PATIENT TNODE, which is created at the initial clinic visit and remains “open” until the patient leaves the medical practice. Because the PATIENT TNODE always spans the temporal duration of all other TNODEs in the network, static information stored within this node is accessible to all TNODEs. The ability to place attributes having temporal duration in one location accessible to all subintervals minimizes data redundancy. Also, it permits interval attributes to be stored within the interval node that represents the proper temporal duration. The inclusion of static data in the PATIENT TNODE allows both static and dynamic data to be maintained within a single, uniform database representation. Separate methods for storing and retrieving static and dynamic attributes are not required.

Each patient has a unique TNET that contains that patient’s data and clinical events. Between consultations, a patient’s temporal network is saved on disk; it is reloaded into ONCOCIN at the next patient visit. Therefore, the temporal network functions both as the patient database and as a representational structure for temporal queries about patient data.

Because open nodes represent events that are currently happening, the set of all open TNODEs defines the current clinical context for the patient. The *leading edge* of the temporal network, depicted as the rightmost nodes with right arrows in Figure 4.4, denotes the most recent clinical events. New TNODEs are added to the leading edge of a temporal network by the ONCOCIN REASONER. The ONCOCIN REASONER contains domain knowledge about the key clinical events that are defined in each protocol. Given both the static knowledge in the protocol and the dynamic information supplied by the user, the ONCOCIN REASONER has sufficient knowledge to determine the beginning or ending of key events in the protocol. When the ONCOCIN REASONER determines that an event has begun, it creates a new open TNODE; when the ONCOCIN REASONER determines that an event has concluded, it closes a previously opened TNODE. Note that the patient’s TNET will always be a reflection of the patient’s clinical course as seen by the ONCOCIN REASONER.

Some TNODEs have a special temporal relationship where one TNODE is temporally contained wholly inside another TNODE. For example, in Figure 4.4, SUBCYCLE TNODEs are confined to the temporal extent of a POCC TNODE. TNODEs with this special relationship are called *parent-child* TNODEs. Parent-child TNODEs represent time intervals that have a domain-specific contextual dependency constraint between the parent

and the child. The term *contextual dependency* refers to the situation in which the event represented by the child TNODE cannot exist without the presence of the event represented by the parent TNODE. In Figure 4.4, if the ONCOCIN REASONER decides to close the CHEMOTHERAPY TNODE, then any child CYCLE TNODE also must be closed. In this example, *contextual dependency* encodes the concept that it makes no sense to say that the patient is still receiving a cycle of chemotherapy when chemotherapy has been terminated. Children TNODEs can close independently of the parent TNODE, whereas a closed parent TNODE will always terminate contextually dependent children TNODEs. Contextual dependency is a concept that appears frequently among the temporal entities in cancer chemotherapy. If a temporal dependency is defined, TNET enforces the parent-child closing property. If no temporal dependencies are defined, TNODEs can be opened and closed independently of the status any other TNODE in the network.

In clinical practice, multiple events happen simultaneously. TNET represents concurrency by permitting TNODEs in separate hierarchies to share concurrent TNODEs. (Figure 4.5). TNODEs in each hierarchy are free to begin new intervals, terminate existing intervals, and enforce temporal dependency constraints within their own hierarchies while sharing TNODEs with other hierarchies. The definition of *contextual dependency* is slightly different for a shared TNODE from that I described in the previous paragraph. Unlike an unshared TNODE, which must close when its parent node is terminated, a shared TNODE terminates only if all parent TNODEs in all hierarchies that share that TNODE are terminated. A shared TNODE also can be closed independently of any of its parents. In all other respects, there is no special handling of shared versus unshared TNODEs.

4.3.2 Physical Implementation

TNET is implemented in a LISP-based object-oriented programming environment called OZONE [70]. An OZONE *object* is a LISP data structure consisting of one or more *spaces*. Each space contains one or more *slots*. A slot contains a *slot name*, a *slot value*, and a *property list* of name-value pairs (Figure 4.6). OZONE supports object-oriented programming features, such as slot-value inheritance, object message sending, and procedural attachments (demon functions) [28]. TNET constructs TNODEs from OZONE objects by adding three TNET-specific spaces—TNET, TCHILDREN, and RESULT spaces (Figure 4.6). In the following sections, I shall describe the structure and function of the

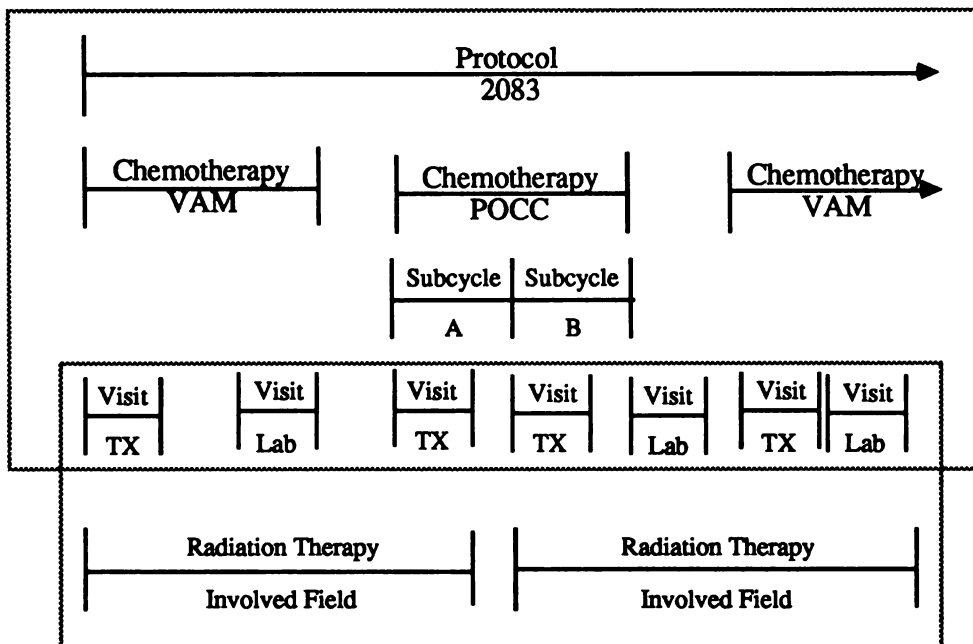


Figure 4.5 Representation of concurrent events in TNET.

This patient is receiving multiple courses of alternating chemotherapy, denoted by the hierarchy of TNODEs enclosed in the upper box. In addition, he simultaneously is receiving repeated courses of radiation therapy, denoted by the hierarchy of TNODEs enclosed in the lower box. Chemotherapy and radiation-therapy events are represented by concurrent, independent TNODEs, whereas visit events are represented by shared TNODEs.

```

(TNET
  (TIMES      ("12-Apr-87" "2-May-87"))
  (TYPE       (CHEMOTHERAPY 5 OBJ.1011 OBJ.1032))
  (SUBTYPE    (POCC 2 OBJ.1011 OBJ.1045))
(TCHILDREN
  (SUBCYCLE   (OBJ.1012 OBJ.1027))
  (SUBCYCLE_A (OBJ.1012 OBJ.1012))
  (SUBCYCLE_B (OBJ.1027 OBJ.1027)))
(VISIT       (OBJ.1013 OBJ.1030))
(TX          (OBJ.1013 OBJ.1030))
(LAB         (OBJ.1022 OBJ.1022))))
(RESET
  (CYCLE#     5)
  (CYCLE.NAME "POCC"))

```

Figure 4.6 The structure of a TNODE object.

TNODEs are OZONE objects that are composed of TNET-specific spaces (TNET, TCHILDREN, and RESULT), slots (TIMES, SUBCYCLE, and CYCLE#), and slot properties (SUBCYCLE_A) and TX). The entries in the TNET space maintain temporal orderings between pairs of sibling objects having the same type and subtype label. The entries in the TCHILDREN space maintain temporal dependencies among parent-child nodes. The entries in the RESULT space store asserted attribute values.

TNODE spaces. In Chapter 7, I shall describe new features that were added to the structure of a TNODE to extend TNET to ETNET.

Each patient in ONCOCIN has a unique TNET that consists of a collection of TNODEs. After each ONCOCIN consultation, a patient's TNET is saved onto a file server for later retrieval prior to the next clinic visit.

4.3.2.1 The TNET Space

The TNET space is responsible for maintaining the temporal ordering of TNODEs having the same type and subtype labels. In Figure 4.4, the TNET space ensures that all

TNODEs of type CHEMOTHERAPY or of type VISIT, subtype TX, are in proper temporal order. The TNET space contains three slots: TIMES, TYPE, and SUBTYPE (Figure 4.6). The TIMES slot contains is a list of two time strings—an opening time and a closing time. All TNODEs contain an opening time, which denotes when the interval event represented by the TNODE started to exist or to happen. A TNODE with a time string for a closing time denotes a *closed* interval, whereas a TNODE with a NIL time string for a closing time denotes an *open* interval.

The TYPE and SUBTYPE slots maintain a doubly linked list of sibling TNODEs. Both slots contain four values: NAME, NUMBER, PREVSIB, and NEXTSIB. In the TYPE slot, NAME contains the TNODE'S type label; in the SUBTYPE slot, this value is the subtype label. NUMBER contains the number of TNODEs of this type or subtype that have been created. PREVSIB (NEXTSIB) contains the name of the TNODE with the same type or subtype label that immediately precedes (follows) this TNODE. The terms *immediately precedes* and *immediately follows* refer to a partial temporal ordering. The *preceding* TNODE must begin either before or at the same time as, and must end either before or at the same time as, the current TNODE. This partial temporal order is imposed on the *following* TNODE also. This restricted definition of *precedes* and *follows* limits the temporal relationships that TNET can capture. This representational limitation in TNET led to the additional extensions in ETNET that I shall discuss in Chapter 7.

4.3.2.2 The TCHILDREN Space

TNODEs that have a parent-child relationship are contextually dependent (see page 103). The TCHILDREN space maintains a list of TNODEs that are contextually dependent on the current node. If no contextual dependency exists, the TCHILDREN space is empty; otherwise, the TCHILDREN space contains one slot for each TNODE type label that is contextually dependent. In each slot is a list of the dependent TNODEs.

The TCHILDREN space encodes the hierarchical nature of a TNET that contains contextual dependency. The TCHILDREN space is used extensively by TQUERY. In a temporal network with concurrency, more than one TNODE may contain the same child TNODE in its TCHILDREN space. In this situation, the child TNODE is shared by both parent TNODEs (see page 104).

4.3.2.3 The RESULT space

The RESULT space performs the database-storage function of TNET. It contains all attribute values asserted by ONCOCIN. Slot names are the attribute names; slot values are the asserted attribute values. Although a TNODE has many *potential* attributes, only those attributes that have asserted values will have slot names created.

Because the ONCOCIN patient record is a legal medical document, the data-storage functions of the RESULT space required special consideration. For example, during a consultation, the user may enter an erroneous value in the patient record. If the error is corrected immediately, ONCOCIN can correct the therapy plan by discarding the erroneous value and reconstructing the recommendation using the correct data. However, if the erroneous data were corrected after the recommended therapy was given, it would be legally improper to rerun ONCOCIN with the corrected data. Nonetheless, for ONCOCIN to generate accurate recommendations in future consultations, the correct values should be used. My solution is to permit old data values to be corrected at any time, but to retain past values, conclusions, recommendations, and actions. A query request will always return only the most recently asserted value for an attribute. In this way, the patient's database can be reconstructed as it was when ONCOCIN constructed the therapy recommendation, but erroneous data can be corrected, and the corrected values can be used in later consultations.

4.4 TQUERY: A Temporal Query Language

TNET provides a mechanism for context-sensitive data retrieval. In our domain, representative clinical contexts are LAST VISIT, LAST CYCLE of CHEMOTHERAPY, and ALL VISITS during the LAST COURSE of XRT. In most cases, the date of the last visit or last course of radiation therapy is not known. TQUERY is a language for specifying data storage and retrieval using clinical contexts rather than calendar dates.

A database operation to a patient's TNET using TQUERY has the general form

(FUNCTION ATTRIBUTE-NAME WHEN)

which reads: "Perform this FUNCTION on the ATTRIBUTE-NAME obtained during WHEN." TQUERY defines the syntax of the WHEN portion of a database operation. It is the WHEN field that encodes the contextual qualifiers that must be satisfied by all attribute values that are supplied to FUNCTION. The WHEN specification uses the temporal

network to select portions of the patient database that satisfy WHEN conditions. By specifying contextual constraints in a WHEN specification, TQUERY ensures that FUNCTION is applied to only those data that were obtained in the proper context.

The WHEN syntax is recursively defined with the following general form:

(INTERVAL-NAME RANGE [WHEN] PNAME PCONDITION)

INTERVAL-NAME is either a TYPE label or a TYPE-SUBTYPE label pair (Table 4.1). It specifies the type of TNODE to be retrieved. A SUBTYPE designator further limits valid TNODEs to those that have the specified type and subtype labels. RANGE denotes which intervals of the proper type are wanted. RANGE may be one of three forms: (1) a distinguished symbol, (2) a single integer, or (3) a pair of integers. Predefined symbolic ranges, such as FIRST, LAST, or CURRENT, restrict the valid TNODE to the first, last or current open TNODE, respectively. Single integer ranges specify which TNODE in a list is valid. Positive integers select TNODEs in temporal sequence (number 1 is earliest, followed by number 2) whereas negative integers select TNODEs in reverse temporal sequence (number -1 is most recent, number -2 in second most recent). The number 0 (zero) is special, denoting the current open TNODE. If no TNODE is currently open, the number 0 returns NIL. A range specified as a pair of integers denotes a list of TNODEs, starting with the TNODE specified by the first integer and ending with the TNODE specified by the second integer. The recursive WHEN clause, discussed in more detail in the next paragraph, provides a method for limiting the scope of the other query arguments (INTERVAL-NAME, RANGE, PNAME, and PTYPE) to only those TNODEs that have already satisfied the recursive WHEN specifications. PNAME and PCONDITION further restrict the desired TNODEs. PNAME without PCONDITION returns TNODEs that have any value for the parameter PNAME. If PCONDITION exists, then the parameter PNAME must exist and its value must satisfy the logical condition specified in PCONDITION.

The recursive WHEN clause restricts the scope of the outer INTERVAL-NAME, RANGE, PNAME, and PCONDITION conditions to that portion of the TNET that satisfies the inner WHEN specification. For example, to retrieve from a patient's TNET all first visits during all courses of VAM chemotherapy, we first restrict the query context to all TNODEs of type CHEMOTHERAPY and subtype VAM, using a recursive WHEN clause, then further restrict these intervals to only first visits, using the outer WHEN fields.

Table 4.1 demonstrates the expressive power of TQUERY by presenting examples with increasing levels of sophistication. In Appendix A, I present the formal BNF specification of TQUERY.

TQUERY is a procedural query language. The order of the nested WHEN specifications influences the results returned and the efficiency with which those results are found in the database. Despite its simplicity, the TQUERY language has proven to be sufficiently powerful in workstation-ONCOCIN to support the construction of complex temporal rules.

4.5 TNET Limitations

TNET embodies several assumptions about the nature of temporal events in ONCOCIN. The most significant assumption is that interesting interval types must be specified in advance. Events that are not known to the reasoning system will not be represented in the temporal network, because it is the ONCOCIN REASONER that is responsible for opening and closing nodes in the temporal network. The definition of an *interesting* interval depends on the temporal model of the problem domain encoded in the ONCOCIN REASONER, which reflects the temporal model defined during the knowledge-engineering process. In a medical environment that had many unexpected or rare events, the ONCOCIN TNET formalism would require each potential interval type to be considered by the ONCOCIN REASONER, to determine whether an interval of this type should be started. A significant amount of processing would be consumed in ruling out the large number of infrequent events.

ONCOCIN deals exclusively with events for which the starting and stopping times are known without ambiguity. Each event's starting and stopping time must be determined precisely. In TNET, absolute timing comparisons are simple numeric operations. Unlike Kohane's TUP program (Section 2.2.1.2.2), the addition of ambiguous or "fuzzy" time cannot be represented adequately in a structure as rigid as is the current form of the temporal network. Although clinical events (such as the onset of a significant drug reaction) do occur between visits, the ONCOCIN problem-solving model does not require precise information regarding the onset of various findings or procedures. In other planning settings, however, encoding temporal uncertainty might be necessary, in which case TNET would be inadequate.

Workstation-ONCOCIN begins a consultation by creating a new visit TNODE

Table 4.1 TQUERY examples with increasing level of complexity.

TQUERY form	Interpretation
(VISIT (1 4))	The first through fourth TNODEs with type label = VISIT.
(VISIT (4 1))	The fourth through first TNODEs. This is the reverse chronological order from the example above.
(VISIT 1) (VISIT (1 1)) (VISIT FIRST)	The first TNODE with type label = VISIT. These three forms are equivalent. The symbolic range FIRST is a macro for 1.
(CYCLE (1 0)) (CYCLE ALL)	The first through current TNODEs with type label = CYCLE. Note the use of zero to denote the current TNODE.
(VISIT (-1 -4))	The most recently closed through fourth most recently closed TNODEs with type = VISIT. Negative numbers denote reverse chronological order.
(VISIT -1) (VISIT (-1 -1)) (VISIT PREVIOUS)	The most recently closed TNODE with type = VISIT. The symbolic range PREVIOUS is a macro for -1.
(VISIT FIRST (CYCLE (-4 -1)))	A recursive query. The inner clause returns the last four closed TNODEs with type label = CYCLE. From each of these TNODEs, the outer clause returns the FIRST TNODE with type label = VISIT.
((VISIT TX) ALL ((CMTX POCC) ALL) WBC (NCOMPARE > \$ 4.5))	A recursive query that uses PNAME and PCONDITION. The inner clause returns all TNODEs with type label = CMTX and subtype label = POCC. From each of these TNODEs, the outer clause returns all visits with type label = VISIT and subtype label = TX in which the value of the attribute WBC exists and it has a value greater than 4.5.

to store the new information provided by the user. This visit TNODE is created at the start of the reasoning process, before ONCOCIN has had a chance to determine whether this clinic visit is the start of a new chemotherapy, the end of a previous subcycle, or a visit during which chemotherapy will not be given. It is therefore initially unclear which superior intervals should be associated with the newly created visit TNODE. By the end of the consultation, these issues have been resolved. TNET is forced to treat the newly created visit TNODE in a manner different from the way it handles all other nodes in the temporal network until it has determined the new TNODEs's proper place in the patient's TNET. A TNODE with this special status, called an *orphan TNODE*, requires special routines to ensure that it is processed correctly even though its proper location within the current temporal network is still uncertain. By the end of the consultation, there is sufficient information to ensure that all orphan TNODEs are incorporated into the temporal network.

The combination of TNET and TQUERY improves data storage and retrieval using date-free, context-sensitive queries. For example, it is both simple and efficient to query for certain visits that occurred during specific chemotherapies. The majority of ONCOCIN's queries are of this form. Occasionally, however, an ONCOCIN rule needs a value obtained a known number of days prior to the current consultation or a known number of weeks after the last chemotherapy. Both TNET and TQUERY handle these situations poorly. TNET must resort to serial comparisons of start and stop dates to find TNODEs that satisfy the calendar date. TQUERY does not provide a language for combining context-free and date-specific queries (for example, "all WBC counts that were obtained 5 days after the start of each cycle of VAM chemotherapy"). Queries of this type are performed using the PNAME and PCONDITION fields. Date-comparison functions are used as the PCONDITION predicate to determine whether a TNODE falls in the right temporal window. This approach hides the nature of the temporal comparison that is being performed in the LISP code that encodes the PCONDITION predicate, which was precisely the same argument used to justify revamping mainframe-ONCOCIN's temporal processing. Fortunately, ONCOCIN rarely requires queries of this form. The problem of defining a language for TNET that efficiently and effectively expresses both context- and date-specific queries remains unsolved.

4.6 Summary

I have discussed several problems in mainframe-ONCOCIN's handling of temporal relationships in reasoning about a patient's clinical status. I presented a structure called the temporal network (TNET) that is interposed between the ONCOCIN reasoning system (the ONCOCIN REASONER) and the patient database. TNET provides a mechanism for marking important temporal-interval events that occur during in a patient's clinical course. Once created, a patient's TNET grows to reflect ongoing clinical events. TNET is used by the ONCOCIN REASONER as a temporal model of the patient's past medical history. TQUERY, permits the ONCOCIN rules to formulate complex data queries that are contextually dependent on temporal events. TQUERY is sufficiently expressive to allow a wide range of temporal specifications to be encoded by the ONCOCIN REASONER. Enhanced by this new structure, workstation-ONCOCIN has a richer temporal model of the patient's past events, with which it can interpret the available data.

CHAPTER 5

The TOPAZ Methodology

Summarizing time-ordered data taken from complex dynamic systems is a difficult task. In the setting of changing expectations, detecting unusual features and abstracting their underlying implications require detailed temporal knowledge. In medicine, insufficient understanding of the interplay between normal and abnormal body processes makes interpreting time-ordered clinical data even more problematic. The natural history of a disease and the alterations caused by therapeutic interventions result in simultaneous interacting influences. A clinical observation cannot be analyzed simply as an isolated number, but rather must be interpreted in light of knowledge about the clinical context that existed when the observation was made. The result is that summarizing a patient's medical record is both knowledge-intensive and context-specific.

In previous chapters, two themes recurred in various forms: (1) an explicit model of the temporal relationships that exist between measurable features and underlying entities constrains the number of meaningful patterns among time-ordered data, and (2) the set of clinical contexts that exist when observations are made alters the assumptions, features, and abstractions that apply to a particular decision. In this chapter, I introduce the methodology used to implement TOPAZ. The exploitation of multiple temporal models and the use of clinical contexts are the two central features. In Chapters 6, 7, and 8, I present the implementation details of TOPAZ as an example of this methodology. In Chapter 9, I

reexamine the methodology and the implementation in light of the past work in temporal reasoning and representation presented in previous chapters.

Most medical knowledge refers to sets of patients with similar characteristics. Phrases such as “most patients with MOPP-sensitive¹ Hodgkins lymphoma show rapid tumor regression after three or four chemotherapy courses” describe the expected clinical course of a hypothetical, *average* patient with a particular medical problem. Because a patient with a given diagnosis has characteristics that distinguish him from the typical patient, each patient’s clinical course will differ from the prototypical expected course. For decisions to be tailored for a specific individual, unique patient features must modify general medical knowledge. Many critical patient characteristics are detected only over the course of treatment. For example, an unusual sensitivity to chemotherapy frequently cannot be identified until one or two courses of treatment have lead to unexpected complications in a certain patient. Early in his clinical course, the patient may show some signs of possible increased sensitivity, but it is only after accumulating confirmatory evidence over time that the physician can make this inference with confidence. After following a patient for some time, an experienced clinician gradually forms an appreciation for the individual’s idiosyncratic response to the illness and treatment. This personal insight develops only after the physician has observed the patient through various phases of disease. Small variations in a patient’s clinical course can have significant immediate and long-term prognostic and therapeutic implications. These subtle differences simply cannot be ignored. A skilled summary of the patient’s medical record should recognize the same patient-specific implications that clinicians derive when they review a patient’s chart. To meet this requirement, we need a method that specializes general medical knowledge based on an individual’s clinical observations.

The TOPAZ methodology stresses three specific tasks (Figure 5.1):

1. *Model-based data interpretation.* An explicit model of the temporal relationships among observations and underlying clinical concepts is required to detect significant temporal features in time-ordered data. A model encodes which temporal patterns are expected (predicted) and which model features, if modified, bring the model predictions closer to the actual observations.

2. *Interval-based data abstraction.* The essence of summarization is combining similar

¹MOPP = mustard, oncovin, procarbazine, and prednisone.

information into a single feature that embodies the larger body of information. In summarizing time-ordered data, intervals can abstract point observations or combine multiple smaller intervals.

3. *Problem-based text generation.* For a summary to be effective, it must highlight the important abstractions and ignore the irrelevant features. For a summary to be convincing, it must include supportive and conflicting evidence for each abstraction that is concluded. Problem-based text generation generates textual descriptions of “interesting” abstractions and provides information on the ability of the model-based data-interpretation to explain the abstraction.

Each task is presented in detail in the following sections. This separation is used only to present the main themes of the methodology. In a specific implementation, the three principle tasks interact. In Chapter 9, I discuss these interactions, and the problems caused by their existence.

5.1 Model-Based Data Interpretation

The TOPAZ methodology begins with an explicit temporal model that encodes general knowledge (Figure 5.1). This model incorporates what is known about the underlying features, processes, and relationships that can cause specific measurements to change over time. The emphasis is on a causal model that relates changing, measurable observations to underlying mechanisms. In medicine, an understanding of causal physiological processes is important for effective clinical problem solving. Therefore, the TOPAZ temporal model encodes physiological knowledge. Like the clinician who revises his opinion as new patient features are observed, TOPAZ modifies its initial temporal model as additional patient-specific measurements are observed. Gradually, the general model becomes patient-specific as new observations change the model’s behavior. The current patient-specific model represents the result of general medical knowledge that has been modified by patient data. TOPAZ derives individualized summary statements by comparing predictions generated by the patient-specific model with those generated by the initial general model. The smooth transition from the type of knowledge typically found in the medical literature (general medical knowledge) to the type of knowledge needed to treat a specific individual optimally (patient-specific knowledge) occurs as new patient data generate a unique pa-

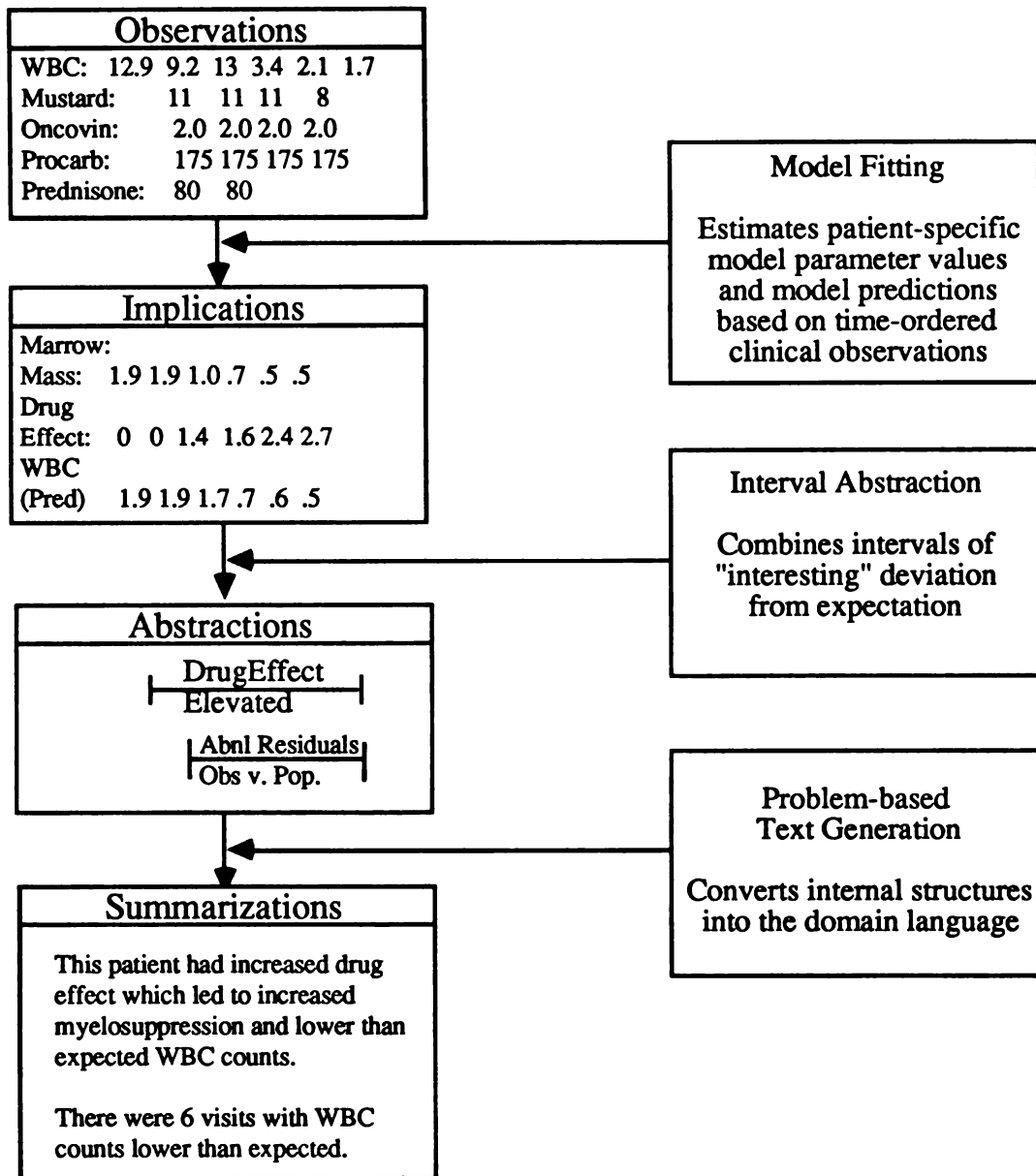


Figure 5.1 Architecture of model-based summarization.

Summarization of time-ordered data is decomposed to three steps: (1) model fitting, which estimates system-specific model features from time-ordered observations using an explicit representation of the temporal relationships among system inputs, outputs, and underlying features; (2) interval abstraction, which aggregates periods of time during which model predictions and system observations show significant deviations from expectations; and (3) text generation, which selects "interesting" abstractions and presents them in the language of the problem domain. Compare this figure with Figure 1.4.

tient model. When the patient has no previous medical history, only general knowledge is available for making initial decisions. As data are obtained, the patient-specific knowledge replaces general medical knowledge, and tailored patient-specific decisions replace general decision-making guidelines.

In Chapter 2, I examined alternative dynamic (temporal) models. A model is a simplified, explicit representation of reality. General domain knowledge describes the entities and relationships that exist in a prototypical instance of a system. Although a model is necessarily only a limited view of reality, a well-constructed model provides considerable leverage in analyzing real-world observations. A dynamic model encodes preexisting knowledge about temporal relationships among entities in the system being represented. A dynamic model expresses the expected changes over time among inputs and outputs. For example, a pharmacokinetic model is a dynamic model that encodes the expected temporal changes in plasma drug levels after the administration of a known amount of drug (Figure 5.2). Interesting dynamic models relate observable phenomenon to unobservable, underlying features. For example, the pharmacokinetic model in Figure 5.2 contains two unobservable features: (1) the volume of drug distribution (V_D) and (2) the elimination rate constant (K_e). With time-ordered measurements, we can use the constraints represented in a dynamic model to make inferences about unobserved phenomena in a specific instance (a patient) of the physiological system (a typical patient) that is represented by the model.

A model encodes general knowledge about a general problem. Each real-world instance has unique characteristics that distinguishes the actual physical system from the modeled system. I use the terms *prototypical model*, *atemporal specific model*, and *temporal specific model* to refer to distinctions among models that are closely related (Table 5.1). A prototypical model represents the entities and relationships that exist in the ideal system rather than those that exist in any particular instance. An atemporal specific model represents a prototypical model that has been altered to reflect important time-invariant features of an instance. A temporal specific model is a prototypical model that has been altered to reflect important time-varying behavior. For example, a model that predicts plasma drug levels after an intravenous injection might initially be based on a prototypical 70-kilogram healthy white male. Additional adjustments to this initial model may occur if the current patient is a 55-kilogram black diabetic female. The model that incorporates the differences in age, gender, health status, and ethnic group is an atemporal patient-specific model (PSM). More model adjustments may occur as measurements of a patient's plasma drug

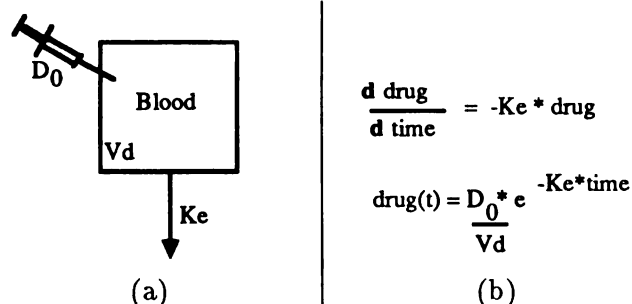


Figure 5.2 A one-compartment drug-elimination model.

Compartment models frequently are used to model physiological processes. Boxes represent hypothetical compartments where mass may enter or leave. Lines represent input and output relationships. This example shows a simplified one-compartment drug-elimination pharmacokinetic model in graphical and mathematical form. Drug enters the compartment directly (such as occurs with the intravenous injection of a drug directly into the blood compartment) and leaves via a single output. The formal model parameters refer to underlying physiologic features, such as the volume of drug distribution (V_D) and the drug-elimination rate (K_e).

levels are made. These changes further individualize the model to the particular patient being studied. The model that incorporates the temporal features is a temporal PSM.

The distinction between atemporal PSMs and temporal PSMs is based on the nature of the observations necessary to reveal the required changes in the model, not on a difference in the model's behavior. In the plasma drug-level model example, the patient's weight would modify a prototypical model into a atemporal PSM.² The effect of this modification would be altered predicted drug levels over time. The atemporal PSM has a different dynamic behavior from that of the prototypical model due to the model alterations caused by a static observation. Conversely, serial plasma levels allow us to infer a patient-specific drug-elimination rate constant (K_e , Figure 5.2). The observations required to calculate or estimate the value of K_e are temporal, but the variable K_e is constant. The distinction between atemporal PSMs and temporal PSMs is useful because it provides a mechanism to distinguish between changes in model behavior due to unusual static patient features from

²In this example, small changes in a patient's weight are considered insignificant relative to the dynamics of the drug model. This assumption of the model may be incorrect in certain clinical contexts. For example, for a patient with renal failure, any change in weight is significant. This point is discussed further in Chapter 9.

Table 5.1 A nomenclature for related patient models.

Model class	Represented knowledge	Incorporated changes
prototypical patient model	general medical knowledge	none
atemporal patient-specific model	patient-specific features	atemporal features
temporal patient-specific model	patient-specific features	temporal trends

changes in model behavior discerned by unusual patient observations over time.

Parameterized models, such as the compartment model in Figure 5.2, constitute a class of models that can be altered to incorporate atemporal and temporal patient-specific findings. The one-compartment drug-elimination model in Figure 5.2 has two parameters, the drug elimination rate constant (K_e) and the volume of drug distribution (V_D). The prototypical model would assign typical or average values to these parameters. I use the term *population-based parameter value* to denote the expected value of a parameter prior to any adjustments due to patient-specific characteristics. A atemporal or temporal PSM would assign new values to the model parameters based on static or temporal observations. The goal of a patient-specific model is to create a model whose predictions more closely match the patient observations than do the predictions produced by the prototypical model. Figure 5.3 illustrates the differences in predictions generated by a prototypical model and a temporal PSM. This figure plots predicted drug levels using the one-compartment drug-elimination model illustrated in Figure 5.2. Figure 5.3 shows two hypothetical plots of this model using two different values of K_e and four sample data values. The dashed-line plot represents the predicted drug levels using the prototypical model; the solid-line plot represents the predicted drug level using the patient-specific model, which has incorporated the four data values. I have simplified this example by allowing only a single parameter (K_e) to change due to the patient-specific measurements. Both models predict the expected drug level at any moment in time following a single drug injection. The patient-specific model, however, more closely predicts the patient data than does the prototypical model.

A general model becomes more specific with the incorporation of new observations, so the specific model better predicts the actual observations (Figure 5.3). A model that encodes knowledge about the underlying structural relationships and more closely predicts the observations is likely to be a more accurate representation of the modeled instance (patient). This improved model is useful for many purposes. In TOPAZ, I use patient observations to

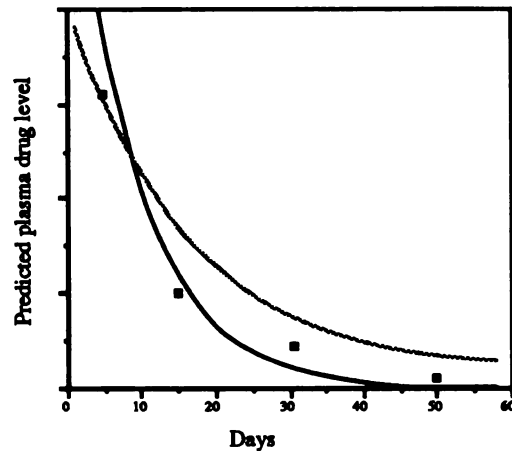


Figure 5.3 Prototypical versus patient-specific model predictions.

This example shows two plots of the model depicted in Figure 5.2. The plots illustrate the model predictions with two values for the drug-elimination-rate constant (K_e). Four patient observations are also shown. The dashed plot displays the prototypical model, which uses the population-based value for K_e ; the solid plot displays the patient-specific model, which uses an estimated value for the patient's K_e . A least-squares regression-fitting procedure was used to estimate the patient-specific K_e from the observations. The parameter V_D was held constant. The temporal PSM better predicts the patient's observations than does the prototypical model.

generate a temporal patient-specific model which, in turn, is used to summarize the patient observations.

The TOPAZ methodology places no restrictions on the formalism used to encode models. The selection of a model-adjusting procedure depends on the formalism used to represent the model. For example, the model that I present in Chapter 6 is a numeric dynamic one; the model-adjusting procedure is least-squares. Had a different modeling formalism been used, a different fitting procedure might have been needed. Irrespective of the representation formalism or of the fitting procedure, the result of model fitting is new model features (parameter values) that reflect changes needed to incorporate specific observations into a more general model.

5.2 Interval-Based Data Abstraction

A second look at Figure 5.3 illustrates the problem addressed in this section. This figure plots the predicted drug level generated by a prototypical model (dashed line) and a patient-specific model (solid line). As expected, the patient-specific model predicts the observations (small boxes) more accurately than does the prototypical model. The patient-specific model now can be used to summarize the four observations. The value of K_e in the patient-specific model is higher than is that of the same parameter in the prototypical model. One possible summary statement merely reports the new K_e value, such as “Based on the series of measured drug levels, this patient’s value of K_e is 0.52.” Although this sentence might be acceptable in some settings, a more useful interpretation is possible. What is missing when we simply present the value of K_e is the notion that, based on our measurements, we believe that this individual eliminates the drug much more quickly than does the typical patient. A clinically meaningful abstraction of the higher K_e value is the concept that an increased elimination constant models more rapid drug elimination. A more insightful summary of the four time-ordered numbers would be the statement “In this patient, the measured drug levels dropped more rapidly than is usual. Based on these observations, it appears that this patient has an increased drug-elimination rate.” This statement removes our ability to calculate predictions over time, but it is clearer than is the actual value of K_e and it still carries the essential clinical implication of the four patient measurements.

Two key points are demonstrated by the summary sentences, “In this patient, the measured drug level dropped more rapidly than is usual. Based on these observations, it appears that this patient has an increased drug-elimination rate.” How do we know that the drug level decreased *more rapidly than is usual* and how do we know that the excessive drop implies *increased drug-elimination rate*? The numeric model in Figure 5.3 simply produces predictions given a set of parameter values. How can a numeric model and its predictions be used to obtain the more insightful abstractions implied by the patient-specific model?

There are two model predictions plotted in Figure 5.3. The prototypical model uses the usual (population-based) patient drug-elimination rate. Therefore, the dashed-line plot represents the *expected* drug level for a typical patient. Large differences between the expected drug level and the actual patient data represent *unusual* or *unexpected* observations. Unusual observations in a patient’s clinical course can be identified as measurements

that deviate significantly from what would be expected or predicted if this individual were a typical patient. Unexpected observations in a patient's clinical course can be identified as measurements that deviate significantly for what would be predicted based on our best current model of this individual.³ The specific model, which incorporates all patient data, also may generate predictions that deviate significantly from the prototypical model predictions. Unlike the actual patient observations, the values predicted by the specific model have largely removed from them the effects of random noise that plague all real-world measurements. In clinical settings, measurement variability can be significant, so large differences between the expected drug level and the actual observation may not reflect a persisting unusual event. To circumvent this problem, TOPAZ uses the set of differences it generates by comparing the prototypical model and the specific model to detect and comment on unusual or unexpected events in the patient's clinical course. Thus, in the example summary sentences, large differences between the expected and the predicted drug levels enable the detection of a drug level that decreases "more rapidly than is usual".

The relationships encoded in the mechanistic model can provide a plausible causal explanation for the differences between expectations and predictions. The model's structural relationships predict that the patient's high drug-elimination rate (K_e) should result in an unusually rapid loss of drug. In the model presented in Figure 5.2, only an increased elimination rate can yield lower-than-expected drug levels. In more complex models, it may not be possible to ascribe an explanation of an unusual finding or trend to a particular model feature or parameter. If multiple processes can potentially contribute to an unusual event, the model structure may not be able to identify a single plausible underlying cause. However, if these processes can cause the unusual event by way of a *common final pathway*, and if the model structure is sufficiently detailed to contain parameters that describe the common final pathway, then it may be possible for the model to detect the unusual final common pathway. In the example summary sentences, an understanding of the model structure and relationships enables the association of lower-than-expected drug levels "due to" an increased drug-elimination rate.

There are additional concepts implied by a model that we cannot capture simply by

³In common conversation, the terms *unusual* and *unexpected* tend to mean the same thing. In this chapter, these two words represent different concepts. An observation may be unusual but not unexpected. For example, if a patient receives twice the normal dose of chemotherapy, his WBC counts will be unusual, but if we have an accurate model of this patient, the counts will not be unexpected. This distinction between unexpected and unusual observations is used in TOPAZ (Chapter 7).

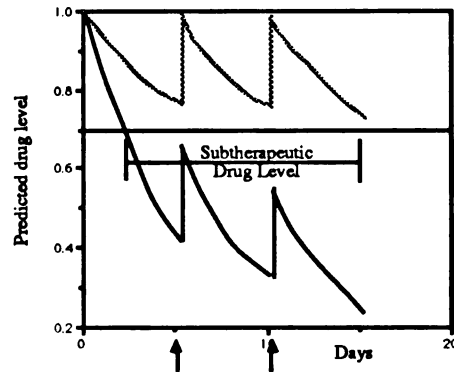


Figure 5.4 Multidose drug levels: prototypical versus patient-specific model predictions.

The two models in Figure 5.3 are used to predict drug levels following multiple drug administrations. Additional drug injections are marked by arrows. The horizontal line denotes the lower bounds for a therapeutic drug level. The dashed plot represents the prototypical model predictions; the solid plot represents the patient-specific model predictions.

comparing model predictions. Figure 5.4, an elaboration of Figure 5.3, illustrates this point. In this figure, the prototypical model and the patient-specific model are used to predict plasma drug levels following multiple drug administrations. The horizontal line represents a hypothetical “therapeutic drug level.” The prototypical model predicts therapeutic drug levels, but the patient-specific model predicts a period of subtherapeutic drug levels. In Figure 5.4, an interval of time denoting the onset and duration of predicted drug levels below the therapeutic value is shown. The presence of low drug levels defines a useful context—that of having subtherapeutic drug levels. This interval abstraction has both clinical meaning and decision-making utility. Although the mechanistic model can provide predicted drug values at any point in time, because models of this class represent time only as a boundless continuous quantity, it would be difficult to construct a mechanistic model that aggregates a set of values into an interval representing a meaningful context, such as periods of subtherapeutic treatment.

In Section 5.1, I proposed using a mechanistic temporal model to derive implications about unobserved concepts from time-ordered measurements. I refer to these models as *mechanistic* models because they relate observations to underlying mechanisms. In this section, I propose a second type of temporal model to extract the concepts and implications

captured by mechanistic temporal models. This other temporal model derives summary concepts and implications that are poorly encoded by mechanistic models. I refer to this class of temporal models as *context* models. Context models have characteristics different from those of mechanistic models, reflecting significant differences in the tasks performed by the two types of model.

I define the term *context* as a period of time during which a meaningful state or situation exists. Because each domain determines what events and situations are meaningful, contexts are domain-specific. A context may describe a moment in time or an interval of time (Figure 5.4). Contexts can occur once and then disappear, can appear intermittently or regularly, or can persist indefinitely once started. As demonstrated in Figure 5.4, I use solid horizontal lines to denote a context, even if that context has negligible duration.

In Chapter 4, I described the ONCOCIN temporal network. This structure models intervals of time during which a patient received various cancer treatments (Figures 4.4 and 4.5). A TNODE denotes when *something interesting* has happened. In ONCOCIN, the set of *interesting* contexts is restricted to temporal events defined by therapeutic event, such as chemotherapy cycles or clinic visits. Because each cancer protocol has different therapeutic contexts, the set of potential ONCOCIN contexts is protocol-specific. Even among patients receiving the same chemotherapy protocol, each patient has a slightly different clinical course. For example, a patient may miss a scheduled appointment, have treatment withheld for 1 week, or be removed from the protocol because of excessive toxicity. Therefore, temporal networks are also patient-specific.

During a prolonged or complex course of treatment, a patient has periods of time during which important nontherapeutic clinical features are present. For example, a patient may show evidence of toxicity that disappears after a change in therapy. The presence of the toxicity, its duration, and its resolution are important events that may be recorded in the patient's medical record whenever they occur. ONCOCIN does not create TNET intervals to represent these events (Chapter 4). In addition to nontherapeutic events, ONCOCIN does not model any inferred contexts, such as increased myelosuppression or rapid drug elimination. For example, ONCOCIN would not generate the interval of subtherapeutic drug level shown in Figure 5.4. ONCOCIN can perform expert therapy planning without any explicit patient-specific model because this knowledge has been implicitly encoded by the clinician who designed the chemotherapy protocol. For example, the ONCOCIN knowledge base contains many rules that determine how to attenuate chemotherapy drug doses for

specific ranges of low WBC counts. Implicit in the selection of the WBC count ranges and in the amount of drug attenuation for each range specified in the protocol is a model of the patient who presents with WBC counts in each range. Although inferred abstractions, such as subtherapeutic drug levels, are unnecessary for ONCOCIN, they are required for an informed abstraction and summarization of temporal events.

Interval-based temporal models, similar to the ONCOCIN temporal network, can represent contextual information. Unlike mechanistic models, which attempt to encode the processes that result in observable behaviors, context models encode the features that relate periods of time to meaningful states. In constructing context models, the model builder is responsible for determining what contexts are *meaningful*. For each modeled context, the model builder also must specify the conditions that define the existence and termination of each type of context.

I use interval-based temporal models to detect unusual clinical events. The context model notes intervals during which significant deviations from the expected clinical course were seen. When an unusual event is detected, the interval-based temporal model encodes the knowledge that attempts to explain the most plausible cause of the unusual events in terms of model parameters and relationships. The context model maps model predictions into clinical implications. It also serves as the bridge between numeric-based methods and symbolic-based methods. Because this model must deal with both numeric and symbolic elements, the knowledge encoded within this model includes both numeric and symbolic concepts.

Figure 5.5 is a copy of Figure 5.3 with two interval abstractions, LOW DRUG LEVEL and ELEVATED DRUG ELIMINATION, superimposed. In this simple example, a context model defines *low* drug level as occurring when there is a significant difference between the drug levels predicted for the typical patient (produced by the prototypical model) and the drug levels predicted for this patient (produced by the patient-specific model).⁴ Once an interval for LOW DRUG LEVEL is created, the context model searches for the most plausible model feature that explains this unusual event. In this simple example, only an elevated drug-elimination rate can cause lower-than-expected drug levels; in more complex cases, however, a number of model features may need to be examined. In this example, the context ELEVATED DRUG ELIMINATION denotes the interval over

⁴Differences between the prototypical model and the actual patient observations are not used at this point in the analysis. This point is discussed in detail in Chapters 7 and 9.

which the patient's observations were sufficiently abnormal to permit the conclusion that the patient's drug-elimination-rate constant was elevated.

5.3 Problem-Based Text Generation

Recounting a patient's clinical course is analogous to telling a highly technical story. There generally is a main plot or theme that embodies a line of reasoning that determines which topics are discussed and in what order these topics are presented. A specific diagnostic or therapeutic problem usually triggers a critical review of the patient's medical record. The items relevant to a concise clinical summary vary depending on the issues raised by the features discovered during the analysis. These unusual or unexpected features narrow the items of interest and focus the analysis of the record. The resulting summary emphasizes findings that contribute to an understanding of the unusual observations and ignores other features that exist in the record. For example, ancestral origin is an important feature in the diagnosis of a particular type of hemolytic anemia, but it is irrelevant in the treatment of lung cancer. The contents of a summary should reflect the needs of the reviewer. Because these needs vary with the set of outstanding problems, the construction of an insightful summary of clinical data is a problem-specific task.

Physicians describe patients to their colleagues using stereotypic presentation styles. There are two key parts to a typical patient presentation: (1) a description of *notable clinical events* that occurred during observation or treatment, and (2) a discussion of the *implications of notable events* for the patient's clinical status. The content of each part is both problem- and context-specific. For example, the level of detail required for effective communication between two experts is very different from the level of detail required for the explanation of a difficult case to a medical student. Among experts, many assumptions are implicitly shared or are explicitly discussed only if some statement is controversial or contested; when an expert teaches students, assumptions and implications are made explicit so that the summary has both informational and educational elements.

If a physician reaches a conclusion from patient data, he usually is able to explain what findings, assumptions, and additional knowledge led him to reach that conclusion. A textual summary should include information that explains how a particular conclusion was derived. One important aspect of this requirement is including known limitations in a particular analysis. A physician can note areas in the record where his statements are not

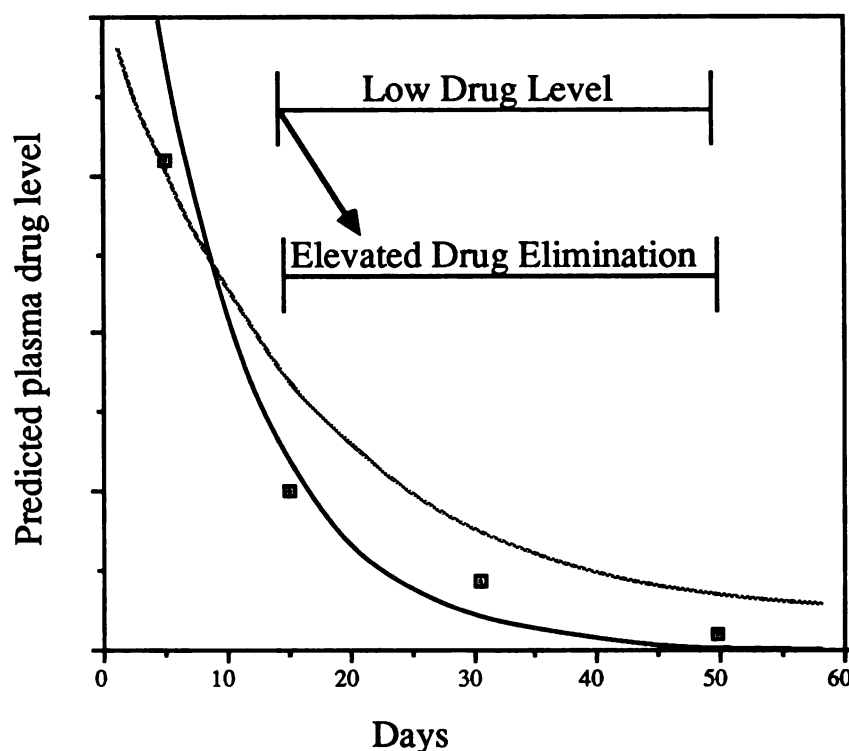


Figure 5.5 Interval abstractions derived from differences in model predictions.

The prototypical model predictions (dashed line), patient-specific model predictions (solid line), and four patient observations in Figure 5.3 are reproduced with two interval abstractions superimposed on the plots. The abstraction **LOW DRUG LEVEL** is created by a large difference between the expected (prototypical model predictions) and patient-specific predictions. The abstraction **ELEVATED DRUG ELIMINATION** is triggered by the creation of the **LOW DRUG LEVEL** context (denoted by the arrow from the **LOW DRUG LEVEL** interval to the **ELEVATED DRUG ELIMINATION** interval), which searches for a plausible explanation for the low drug levels. In this simple example, only an elevated drug-elimination-rate constant can be responsible for the unusual finding. The interval **ELEVATED DRUG ELIMINATION** does not represent the length of time during which drug elimination was elevated; rather, it represents the length of time during which sufficient evidence existed to make this model feature be the most likely causal explanation of the abnormal finding.

well supported or even are contradicted by the data. A computer-based analysis must be able to report features in the final summary that were based on weak or potentially incorrect inferences. Even among experts, the physician can decide whether sufficient evidence still exists to accept the conclusions, even if the data used to justify that conclusion are not strongly supportive.

A useful textual summary must embody at least three essential elements:

1. The text must follow the stylistic conventions that are expected by the reader of the summary
2. The text must contain sufficient detail to support stated conclusions, as appropriate for the expertise of the reader
3. The text must identify periods in which the analysis may be incorrect or tenuous, so that the reader can determine whether the conclusions should be accepted.

The interval abstractions discussed in Section 5.2 denote the existence of unusual or unexpected events in the patient's medical record. These intervals form the basis for constructing a patient-specific summary that is focused on only those problems actually encountered during the clinical course. Although enumerating the unusual or unexpected findings is useful for focusing attention on key problems, a more interesting analysis relates these findings to the probable underlying process that caused the unusual finding to occur. One aberrant underlying process may trigger multiple unusual events. For example, a patient with a marked sensitivity to myelosuppression from chemotherapeutic drugs will exhibit multiple episodes of depressed WBC counts after each chemotherapy cycle, with normal counts between treatments. In this setting, the occurrence of multiple periods of myelosuppression is interesting; but for therapeutic decision making, the single underlying cause—excessive drug sensitivity—is more important. The text-generation system must be able to interpret the interval abstractions generated by the context models (Section 5.2), to determine their relevance to the overall structure of the summary, and to include or exclude specific comments if the findings confirm or deny relevant concepts. The text-generation system must be flexible enough to accommodate records of patients with vastly different clinical courses, but structured enough to impose the typical stylistic features expected by the clinical reader.

To construct text that embodies the expected form and content of an insightful clinical summary, we must have a third model, called a *presentation model* [83]. The presentation model is constructed from two presentation schemas: (1) a presentation schema of the general organizational structure of a clinical summary (the *organizational schema*) and (2) a presentation schema of the specific structure of a clinical argument (the *rhetorical schema*). The organizational schema encodes the overall flow of topics that typically appear in clinical summaries. The rhetorical schema encodes the details of constructing a persuasive clinical argument that presents and supports clinical interpretations and inferences. The organizational schema provides the global structure in the clinical summary, whereas the rhetorical schema provides the logical structure of the summary contents.

As the number of findings and conclusions increases, the number of comments that *potentially* could be created by the system increases. Not all comments that could be made need to appear in all summaries. Some inferences may be so critical that, if they are concluded, a comment must always appear in the final output. Other comments may be selected or bypassed, depending on the clinical setting or the user's demands. One reader may want only the system conclusions, whereas another reader may demand to see the supporting evidence for all system statements. Expert readers can be assumed to understand the problem domain more completely than can nonexpert readers. Explicit explanations of assumptions that novices require will be unnecessary and irritating to an expert. An ideal interactive presentation model would change the level of summary-text output dynamically in response to real-time user requests for additional information about system conclusions or inference assumptions.

5.4 Model-Based Summarization

In Section 5.1, I argued that a detailed mechanistic model is required for making inferences about key unobserved physiological processes occurring in a patient. Inferences are derived from differences between the patient-specific model predictions and the expected clinical course. The mechanistic model encodes detailed knowledge about the temporal relationships that the domain expert believes exist in the real-world system. In TOPAZ, this model represents the physiologic and therapeutic processes and their effects on bodily functions. Despite the detailed relationships encoded in the mechanistic model, this extensive information appears to be discarded in later abstractions (Section 5.2) when the detailed

model predictions are aggregated into clinically relevant summary intervals (contexts).

What information does the patient-specific mechanistic model provide that the context model cannot derive directly from the observations? The mechanistic model encodes explicit physical constraints that must exist among inputs, outputs, and underlying processes. This initial set of constraints embodies substantial prior knowledge about how simultaneous bodily processes evolve, combine, and interact. These constraints provide the knowledge to derive logically and physically consistent implications from patient observations. With an explicit mechanistic model, basic physical and physiological principles cannot be violated, even if patient data are extremely unusual.

What do we require to make inferences directly from the data without an intervening mechanistic model? Without an explicit representation of the relationships among inputs, output, and underlying processes, each observation must be associated with all the potential implications that could have led to the observed value. This is precisely what was done in Downs' summarization program (Section 3.3.3) and in IDEFIX (Section 3.3.4). With this approach, some method to combine a temporal series of observations is required. One possibility is to form the intersection of the potential implications associated with each observation. As more observations are obtained, the intersection of potential implications aggregates a series of measurements. A second approach is to define implications for pairs, triples, and larger series of observations. In medical settings, there are many simultaneous, interacting processes. Enumerating all the possible implications for a single observation quickly becomes unwieldy. For larger sets of observations, the task is impossible. In most cases, the approach of enumerating all possible implications from an observation is merely an attempt to construct an implicit model of the constraints explicitly imposed in the relationships encoded in a mechanistic model. However, the potential for logical error and incompleteness and the lack of representational clarity makes the enumeration approach unwise.

The problems associated with the ill-advised effort to draw inferences directly from observations without an intervening mechanistic model are illustrated in OBOE (Section 3.4.1). In this program, context-specific implications were associated with ranges of observations that converted patient measurements directly into simple clinical conclusions. Each node in an augmented transition network (ATN) represented a *complete clinical situation*. A complete clinical situation was the combination of all concurrent contexts that could have existed at a particular moment during a patient's clinical course. A complete

clinical situation defined all the simultaneous processes and interactions that could influence the interpretation of an observation. Once a complete clinical situation was defined, ranges of observations were mapped into implications that were consistent with the clinical setting. Because each node in the ATN had to represent a complete clinical situation, the number of ATN nodes in OBOE grew exponentially as the number of possible clinical contexts that could interact and combine with other contexts to define a new clinical situation increased. Both augmenting and refining the OBOE knowledge base quickly became impossible. Particularly difficult was ensuring the consistency of inferences that could be permitted when multiple contexts were combined. For this reason, the OBOE approach was abandoned.

The patient-specific model need not be completely discarded once interval abstractions have been generated. The mechanistic model can provide interpolated predictions about events occurring between observed measurements. These interpolated predictions can provide insights about events that probably occurred but were not observed. The mechanistic model can also be used to make predictions about future events. For example, the patient-specific model can simulate the future clinical course that would be expected if the patient were to receive a potential treatment plan. When used for this purpose, the patient model can serve an important role in evaluating the expected patient-specific outcomes of competing therapy plans. These additional uses of the patient model are not required to generate TOPAZ summaries, but they are a valuable extra benefit that the physiological model can provide to other complex decision-making tasks.

5.5 The TOPAZ Implementation

In this chapter, I have presented a three-step methodology for analyzing and summarizing time-ordered medical data. A mechanistic model of the underlying system structure yields a patient-specific model from actual observations. Differences in the predictions generated by patient-specific model from the predictions generated by the population-based model define “unusual” clinical events. An interval-based model abstracts the patient’s clinical course by aggregating intervals of unusual model predictions. The creation of an “unusual” event interval triggers an examination of the structural features of the patient-specific model for plausible explanations for the unusual findings. A presentation model represents the structural and logical features of a clinical presentation so that each abstraction discussed in the summary is presented in the proper clinical context (state).

In Chapter 1, the three dissertation hypotheses were presented:

1. The diversity of temporal knowledge that is used in complex temporal problem-solving demands multiple representational and inferential formalisms that work cooperatively
2. Explicit models that relate observations to underlying concepts provide a mechanism to interpret, explain, and predict temporal patterns
3. Context-sensitive reasoning limits the size of the problem space by constraining the number of features or relationships that potentially are relevant to the current reasoning task.

The TOPAZ methodology uses three models to encode different temporal aspects of summarizing time-ordered clinical data. The structural model captures the continuous, process-oriented nature of time; the interval-based model captures the discrete, event-oriented nature of time; and the presentation model captures the state-based nature of time. The three TOPAZ models encode markedly different temporal perspectives, yet each perspective is required for the TOPAZ task. No single formalism would be sufficiently flexible to encode the concepts embodied in the other models.

The explicit structural model is essential to the TOPAZ methodology. It is used to construct the patient-specific model from time-ordered observations, to define the concept of an “unusual” observations, and to provide a plausible explanation for the unusual observations. Although prediction is not an essential task for data summarization, the patient-specific model generated by fitting observations to a model can be used to examine potential future clinical events.

The use of context to limit the size of reasoning is developed in Chapter 7. A simple use of context-specific reasoning was demonstrated in Figure 5.5. In this example, the creation of the interval LOW DRUG LEVEL triggered the search for a potential explanatory cause, which resulted in the creation of ELEVATED DRUG EFFECT. Without the enabling context of LOW DRUG LEVEL, the search for ELEVATED DRUG EFFECT would not occur. Not all model features can potentially contribute to an unusually low predicted drug level. Context-specific reasoning is used to limit the search to only those model features that potentially could explain the lower-than-usual predicted drug levels.

TOPAZ is a computer program that constructs textual summaries of context-sensitive, time-ordered data. Specifically, TOPAZ summarizes the hematologic status of

patients receiving MOPP chemotherapy for Hodgkins' lymphoma (Figure 5.6, Figure 5.7). To produce the final text, TOPAZ analyzes time-ordered clinical data for unusual or remarkable events, abstracts the clinically relevant implications from these events, and generates realistic prose that describes the patient's clinical course. In Chapters 6, 7, and 8, I shall show how the three tasks that comprise the TOPAZ methodology (model-based data interpretation, interval-based data abstraction, and problem-based text generation) are implemented in the TOPAZ program. The key themes of multiple temporal models, an explicit structural model, and context-specific reasoning are the centerpieces of the TOPAZ program.

Throughout the next three chapters, I shall use the patient record shown in Figures 5.6 and 5.7 in all examples. As I proceed through the three key phases in TOPAZ, I shall display the intermediate steps leading to the formation of this summary text. Additional patient records, intermediate outputs, and final summary texts are presented in Appendix C.

Upper Nodes	2		2		1	1		1	1	
Lower Nodes	1		1		1	1		1	1	
Spleen										
Splenectomy										
Liver Bx										
Marrow Bx	1									
LAG	2					2		2		
IVP										
CXR	6					2		2		
Bone Scan										
CT Scan										
OVERALL	6			3	3	2		2	2	
Karnofsky	100			100	100	100		100	100	
Weight	90.7		89.5	88.2	87.4	89.1		87.3	88.5	84
WBC X 1000	7.7	7.9	6.9	4.8	3.1	4.4	3.3	4.7	5.3	5.0
PCV	40.9	40.4	37.1	36.3	32.2	32.6	28.2	27.7	24.4	29.9
Hemoglobin	13.1	12.8	12.2	11.8	10.6	10.5	9.4	8.8	9.4	9.2
Platelets	574	329	368	282	305	120	112	193	150	106
BSA (m ²)		2.11								
Chemo Name		MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP
Cycle #		1	1	2	2	3	3	4	4	5
Subcycle		A	B	A	B	A	B	A	B	A
Visit Type	LAB	TX	TX	TX	TX	TX	TX	TX	TX	TX
Mustard		12	12	12	12	10	8	8	8	8
Vincristine		2.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0	1.0
Procarbazine		200	200	200	100	75	75	75	75	75
Prednisone										
Port										
RTx Start Date										
RTx End Date										
Cum. # Fraction										
Cum. Max. Dose										
Number Rads										
RT Toxicity										
Day	19	26	4	30	6	28	4	9	16	8
Month	Oct	Oct	Nov	Nov	Dec	Dec	Jan	Feb	Feb	Mar
Year	83	83	83	83	83	83	84	84	84	84

Figure 5.6 The ONCOCIN record of a patient who has received MOPP chemotherapy.

TOPAZ summarizes the medical records of ONCOCIN patients who have received MOPP chemotherapy. This is the record of the patient whose summary appears in Figure 5.7. The complete medical record includes two additional pages of patient data recorded in flowsheet format, as well as handwritten physician notes. Only those data recorded in the flowsheet are found in the ONCOCIN patient database. TOPAZ uses only the WBC counts, chemotherapy dosages, and clinic-visit dates to construct a summary of the patient's hematologic status during treatment.

Patient Identifying Data:

This is a 25 y/o male who presented with clinical stage III-S-A, pathological stage III-S-A Hodgkins' lymphoma. The patient's medical record begins on 19 October 1983 and ends on 24 May 1984. He was assigned to MOPP chemotherapy and received 7 treatment cycles.

Analysis of Clinical Course:

CAUTION: This analysis is based on models of the effects of MOPP chemotherapy on bone marrow and WBC kinetics to derive inferences about unobserved physiological processes. Model mis-specification and/or measurement inaccuracies can result in serious errors in the conclusions derived from these models. In the analysis that follows, conclusions that are based on periods in the record when the model fit appears to be poor are noted. Please take these limitations into account while reading this analysis.

Based on our analysis of the entire clinical course, the best fit of our model to the data suggests the following key patient-specific physiological parameter values:

1. A steady-state bone marrow mass that is approximately 0.6 times the population value. This is considered a moderate reduction in steady-state bone marrow mass.
2. A duration of drug effect that is approximately 2.2 times the population value. This is considered a moderate increase in the duration of drug effect.
3. A sensitivity to drug effect that is approximately 1.4 times the population value. This is considered a mild increase in the sensitivity to drug effect.

Based on differences between population-based predictions (the expected clinical course) and the patient-specific model predictions (the modeled clinical course), the patient had 2 remarkable clinical events. For each event, we provide a description of the event, a probable cause for the event, and an analysis of the ability of the patient-specific model to predict the actual observations during this event.

1. A series of 2 visits prior to receiving chemotherapy, starting on 19 October 1983 and ending on 26 October 1983, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 26 October 1983 (63% of the expected value).

The patient's estimated steady-state marrow mass was lower than expected causing a lower than expected pre-treatment WBC count. The model cannot explain the underlying etiology of a lower than expected steady state marrow mass.

There were no periods of poor model fit during this remarkable clinical event.

2. A series of 13 visits after starting chemotherapy, starting on 30 November 1983 and ending on 24 May 1984, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 28 December 1983 (64% of the expected value).

After adjusting for the patient's steady-state bone marrow mass, the two drug-related parameters, duration of drug effect and drug sensitivity, jointly caused an increased drug effect which led to increased myelosuppression and lower than expected WBC counts. The increased drug effect had a peak myelosuppressive effect 13.1 times greater than expected on 19 April 1984. The accumulation of drug due to the prolonged duration of drug effect was responsible for a 9.4 fold increase in drug effect. The increased sensitivity to drug was responsible for a 1.4 fold increase in drug effect.

Out of the remaining 13 visits during the period, there were 5 episodes of nonsystematic model versus observation misfitting (6 December 1983, 4 January 1984, 3 April 1984, 26 April 1984, and 24 May 1984). Because these model misfits were not sustained for longer than one observation, we feel they do not alter the parameter estimates derived over this period.

There were no episodes of model misfitting that occurred during the unremarkable periods in the patient's clinical course.

Figure 5.7 TOPAZ-generated summary.

TOPAZ accepts the ONCOCIN medical record of a patient who has received MOPP chemotherapy and produces a textual summary of the hematologic status of that patient during treatment. This is the summary generated by TOPAZ for the patient record shown in Figure 5.6. This patient record is used in all examples in later chapters. This summary and others appear in normal font in Appendix C.

CHAPTER 6

A Process Model of Granulocyte Kinetics and Chemotherapy

The intended action of cancer chemotherapy is to destroy malignant cells selectively. Unfortunately, this goal is rarely attained. Most chemotherapy drugs have significant side effects caused by the undesired destruction of normal cells. A major source of concern in MOPP chemotherapy is the destruction of granulocytic white blood cells (WBCs). A patient receiving MOPP is expected to show some drop in his granulocyte counts approximately 7 to 10 days after starting drugs. Three weeks later, the granulocyte count should be near normal. A physician using myelosuppressive drugs closely monitors the peripheral granulocyte count for any sign of excessive cell destruction. Frequently, the early signs of trouble are subtle. Later, when the patient's unusual response becomes more apparent, the physician's only remaining option is to delay treatment. Numerous clinical studies have shown MOPP chemotherapy to be most effective in curing patients with Hodgkins' lymphoma when the patient receives the full recommended dosages, but continuing therapy in light of toxic reactions can be life-threatening. The treating physician, therefore, must search diligently for early warning signs of myelosuppression that he can correct with small changes in drug dosages before excessive toxicity forces him to terminate the patient's treatment early.

The patient's bone marrow is the key physiological site where excessive destruction causes serious clinical problems. Because we rarely are able to observe the bone marrow directly, the peripheral blood counts are a critical indirect measure of the functional status of this organ. Even though a single granulocyte value has considerable clinical implications of its own (for example, as an indicator of the risk for infection), it has additional decision-making value as a reflection of bone-marrow status. Some representative temporal patterns in WBC counts following MOPP chemotherapy that imply a clinically worrisome bone-marrow status are (1) a WBC count that drops precipitously, (2) a WBC count that reaches a dangerously low value, (3) a count that returns to the pretreatment level only after a prolonged period of time, and (4) a WBC count that never returns to the pretreatment level. There are many possible reasons why a patient may exhibit an unusual response to chemotherapy, such as an unusual sensitivity to the myelosuppressive effects of drug, a prolonged duration of drug action, or a permanent, irreversible loss of granulocyte-production capacity. Both the timing and the value of an observation provide insights into these clinically relevant distinctions. In some cases, patients have markedly abnormal values following the first course of therapy; in most cases, however, they demonstrate a problem only over the course of multiple cycles of chemotherapy. Interpreting a sequence of granulocyte counts over time during repeating courses of chemotherapy is essential. Note that the clinical context changes as the patient receives successive courses of drug. The effects of past treatments may persist into the next drug administration, the patient may receive other treatments such as radiotherapy, and treatment delays may change the timing of treatment while the patient is being observed.

Because of the central role of the bone-marrow status in limiting the delivery of important chemotherapeutic agents and in preventing the use of many alternative treatment modalities, I have focused on capturing the temporal implications of a series of time-ordered granulocyte counts for the underlying status of a patient's bone marrow. The implications that I wish to encode include (1) high sensitivity to myelosuppressive drug effects, (2) prolonged duration of myelosuppressive drug action, and (3) irreversible bone-marrow damage. In the following sections, I describe a physiological model constructed to encode these concepts.

6.1 A Numeric Process Model of the Bone Marrow

Nearly all granulocytic (WBC) production occurs in the bone marrow. Granulocytes are produced by omnipotent marrow stem cells that divide continuously to generate new immature granulocytic precursors. These immature cells, also found in the bone marrow, mature over a number of days before they are released into the peripheral bloodstream. Approximately 5% of the immature granulocytes are destroyed in the bone marrow (ineffective hematopoiesis). Once released into the bloodstream, mature granulocytes remain in the circulation until they are removed in the spleen. Individual patients differ in the number of stem cells producing granulocytes, the amount of ineffective hematopoiesis that occurs in the bone marrow, the length of time required for granulocytes to mature in the marrow, and the length of time mature granulocytes circulate in the peripheral bloodstream. Patients with Hodgkins' lymphoma differ significantly on these physiological features from normal individuals. Hodgkins' lymphoma seems to cause a state of hypercellular granulocyte production. Most patients with Hodgkins' lymphoma who do not have excessive replacement of normal marrow cells with lymphoma have a hypercellular bone marrow on marrow biopies, increased ineffective hematopoiesis, and elevated peripheral WBC counts. The main myelosuppressive effects of MOPP drugs are to increase the destruction of immature granulocytes in the bone marrow and to destroy the number of omnipotent marrow stem cells that produce the initial immature granulocyte.

Using standard techniques of compartment models, I have constructed a parameterized numeric model of the physiological processes in the production, maturation, and destruction of granulocytes (Figure 6.1). I call this model the *bone-marrow model*. Compartment models are used frequently to model physiological processes [2,19]. A compartment model has three main components: (1) one or more hypothetical compartments, (2) input and output connections between compartments, and (3) input and output connections between compartments and the outside world. Connected compartments transfer either mass or information—the difference being that mass is conserved in all transfers, but information is not. Compartment models are used extensively in drug-pharmacokinetics research [45]. The compartment model methodology readily adapts to the bone-marrow problem.

The graphical representation of the TOPAZ bone-marrow model in Figure 6.1 contains all the main features of granulocyte development that I described previously. The bone-marrow compartment (*marrow*), representing immature granulocytes in the marrow,

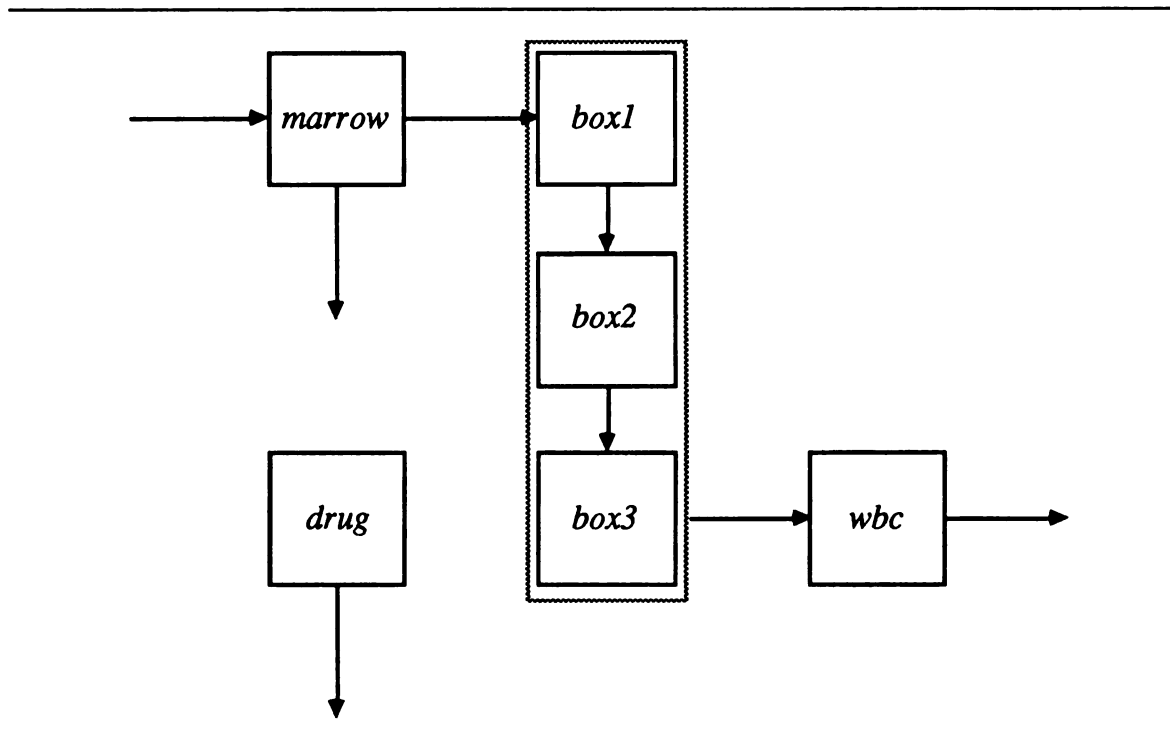


Figure 6.1 Graphical representation of the TOPAZ bone-marrow model.

A compartment model is constructed from hypothetical compartments and transfers between compartments. Boxes represent compartments; lines represent transfers. This is the bone-marrow model used in TOPAZ. The meanings of the model-state variables (*marrow*, *box1*, *box2*, *box3*, *wbc*, and *drug*) are given in Table 6.3.

is replenished by an exogenous source of replicating cells, which represents the omnipotent stem cells. Two losses occur from the bone-marrow compartment. One loss enters a chain of compartments, called the *delay chain*, which represents the maturation phase of granulocytic development. The three compartments that constitute the delay chain (*box1*, *box2*, and *box3*) have no losses except into the next delay compartment. This chain effectively delays and spreads out the effects of granulocytic production and MOPP toxicity as they propagate from the bone-marrow compartment to the peripheral-granulocyte (WBC) compartment (*wbc*). A second loss from the bone-marrow compartment represents the destruction of cells within the bone marrow from both ineffective hematopoiesis and drug toxicity. Myelosuppression from chemotherapy is modeled as an additional increment to the second marrow-loss rate. This increased loss rate is a function of the amount of drug effect remaining in the patient (*drug*) and the sensitivity of this patient to the drug. The WBC compartment, representing the peripheral granulocytes, is the compartment from which observations (WBC counts) are drawn. A loss that is not sensitive to the presence of drug exits from the WBC compartment, which represents the normal turnover of peripheral granulocytes. Additional destruction of peripheral granulocytes by chemotherapeutic agents is not incorporated in this model.

Although MOPP chemotherapy comprises four distinct drugs, each having a specific rate of metabolism and a myelosuppressive potency, the drugs are combined and modeled as one *pseudodrug*. I model the decay of the *drugeffect*, rather than the decay of the actual drug metabolites. This simplification is justified because the two main myelosuppressive drugs in MOPP chemotherapy, nitrogen mustard and procarbazine, have roughly similar acute myelosuppressive effects. This simplification may not be a plausible assumption for other MOPP toxicities or in other cancer-chemotherapy combinations. To model different agents as a single pseudodrug, I normalize each drug dose by that drug's body-surface-area adjusted standard dose, and then multiply that adjusted dose by the drug's relative myelosuppressive potency (Table 6.1). This adjustment is done for all four MOPP drugs, and the values are added to derive a drug-effect level. The relative potency values, which range from 0 to 1, are obtained from clinical experts. A single compartment with a single loss models the drug-effect decay (Figure 6.1). New drug dosages are added directly into the drug compartment. This model neither represents drug-adsorption kinetics nor conserves mass among compartments. The flows among compartments are "information" quantities, which need not be conserved.

Table 6.1 Relative myelosuppressive potency of MOPP drugs.

Drug name	Relative potency
nitrogen mustard	1.0
vincristine	0.1
procarbazine	0.9
prednisone	0.0

I use the general framework of state-based models to describe the mathematical formulation of the TOPAZ bone-marrow model. A state-based model represents a system as a set of system states. A dynamic system is modeled by equations that represent changes among states over time. In addition to states and equations, state-based models have parameters that alter the relationships among system states. Parameters may change over time or be constants. A complete description of a dynamic system using a state-based model requires the specification of the equations that describe state changes (called the *structural model*), initial values for all modeled states and parameters, and all new inputs that enter into the system. After these elements of the model have been specified, a state-based model will produce predicted values for all model states and (time-varying) parameters. I use the following notational conventions to present the TOPAZ bone-marrow model:

1. Lower-case italic symbols denote scalars (e.g., x, y)
2. Lower-case boldface italic symbols denote vectors (e.g., $\mathbf{z}, \boldsymbol{\phi}_0$)
3. Upper-case boldface italic symbols denote matrices (e.g., $\mathbf{B}, \mathbf{C}$)
4. q_i denotes the i th scalar element in the vector $\mathbf{q}$
5. t_i, t_j denote times such that t_i precedes t_j if and only if $i < j$
6. $r_{(i)}$ is a notational shorthand for $r(t_i)$, for all scalar, vector, and matrix symbols r
7. $\dot{s}$ is a notational shorthand for $\partial s / \partial t$ for all scalar, vector, and matrix symbols s

The general form of the state-space equations that I will use to describe the TOPAZ bone marrow model is:

$$\dot{\mathbf{z}} = \mathbf{a}(\phi, \mathbf{z}) + \mathbf{i} \quad (6.1)$$

$$\dot{\phi} = \mathbf{B}\phi + \zeta \quad (6.2)$$

$$\mathbf{y} = \mathbf{z}'\mathbf{C}\psi + \epsilon \quad (6.3)$$

In Equation 6.1, $\mathbf{z}$ is the system state vector. This equation describes the change in the system state vector ($\dot{\mathbf{z}}$) over time as a function of $\mathbf{a}$ and $\mathbf{i}$. The vector $\mathbf{a}$ is the state equations that contains the equations that describe the relationships among system states. In the case of the TOPAZ bone-marrow model, the state equations in $\mathbf{a}$ are functions of model parameters (ϕ) and system states ($\mathbf{z}$). The vector $\mathbf{i}$ represents new input rates that enter the system and that may change system states.

Equation 6.2 describes the change in model parameters (ϕ) over time. The matrix $\mathbf{B}$ describes any functional relationships among model parameters. In the bone-marrow model, $\mathbf{B} = \mathbf{0}$, so Equation 6.2 reduces to $\dot{\phi} = \zeta$. The vector ζ is a vector of random variables. With ζ , we can model random changes in parameter values over time.

Equation 6.3 describes the model predictions ($\mathbf{y}$). The vector $\mathbf{z}'$ is the transpose of the state vector, $\mathbf{C}$ is a design matrix (contains only zeros and ones), ψ is the vector $(1, \phi)'$, and ϵ is a vector of random variables. The term $\mathbf{z}'\mathbf{C}\psi$ describes the relationship among system states and observations. In some settings, we can observe a system state directly; in other cases, we have access only to indirect measures of system states. In TOPAZ, we observe a system state (wbc) when we measure a peripheral WBC count. With the vector ϵ , we can model random errors in the observed values.

In the TOPAZ bone-marrow model, the random variables, ζ and ϵ , are assigned the following properties:

$$E(\epsilon) = 0; \text{var}(\epsilon) = (.06 * \mathbf{z}'\mathbf{C}\psi)^2 \quad (6.4)$$

$$\epsilon_i, \epsilon_j \text{ independent for } i \neq j \quad (6.5)$$

$$E(\zeta) = \mathbf{0}; \text{var}(\zeta) = \mathbf{\Gamma} \quad (6.6)$$

$$\zeta_i, \zeta_j \text{ independent for } i \neq j \quad (6.7)$$

$$\zeta_i, \epsilon_j \text{ independent for all } i, j \quad (6.8)$$

Equation 6.4 states that the errors in observations have an expected value of 0 and

a variance proportional to the square of the predicted value. Equations 6.5, 6.7, and 6.8 state that all errors are independent. Equation 6.6 states that the random effects in time-changing parameters have an expected value of 0 and a variance specified by the matrix Γ .

The initial conditions are $\mathbf{z}(0) = \mathbf{z}_0$, assumed known. The model parameters (ϕ) are random variables drawn from a population of patients with similar clinical characteristics. This property is described by the equation:

$$\phi = \phi_0 + \eta \quad (6.9)$$

In this equation, ϕ_0 represents the population mean parameter value. Each patient has unique parameter values that may differ from the population mean by η . Assuming that the expected value of η is 0 ($E(\eta) = \mathbf{0}$) and $cov(\eta) = \Omega$,

$$E(\phi) = \phi_0 \quad (6.10)$$

$$var(\phi) = \Omega \quad (6.11)$$

$$\Omega = diag(0.3 * \phi_0)^2 \quad (6.12)$$

Equation 6.10 states that the expected value for the initial parameter values is the population value. That is, in the absence of any patient-specific observations, I use the population parameter values in the model.

In TOPAZ, the parameter covariance matrix Ω is defined as $diag(0.3 * \phi_0)^2$. In addition, TOPAZ assumes that each patient has stationary parameter values ($\dot{\phi} = 0$). With this additional assumption,

$$\Gamma = 0 \quad (6.13)$$

In Section 6.4, I discuss the possibility of time-varying patient parameters where Equation 6.13 is likely to be incorrect.

In Section 6.2, I present the features of the TOPAZ structural model (the elements of the vector $\mathbf{a}$ in Equation 6.1), the model parameters (the elements of the vector ϕ in Equation 6.2), and the initial model states (the elements of the vector $\mathbf{z}_0$). In Section 6.3, I present the fitting procedures that provide estimates of the model parameters, ϕ , from patient-specific observations.

6.2 The Structural Model

The TOPAZ bone-marrow model is a set of modeled states whose behavior over time is described by state equations and parameter values (Equations 6.1 through 6.3). In the following sections, I shall decompose the description of the TOPAZ bone-marrow model into a description of the model state equations (Section 6.2.1), the model parameters (Section 6.2.2), and the model states (Section 6.2.3). The goal of these sections is to describe in a formal mathematical system the medical knowledge and intuition about bone-marrow kinetics and chemotherapy depicted informally in Figure 6.1.

6.2.1 Model Equations

Predicted changes in model state variable values occur due to constraints among model elements and new external inputs (Equation 6.1). In TOPAZ, a system of six nonlinear equations (the elements of the vector $\mathbf{a}$) describes the dynamic constraints among model state variables. Figure 6.2 presents the model state equations that encode the temporal relations implied by the boxes and arrows in Figure 6.1. Equations 6.14 through 6.18 encode the bone-marrow, maturation-delay-chain, and peripheral-WBC compartments; Equation 6.19 encodes the drug-effect compartment. The myelosuppressive effect of chemotherapy is represented in Equation 6.14 as the additional drug-dependent fractional decay term, $d_{sensiv} * drug$.

The TOPAZ physiological model is implemented in FORTRAN using a fifth-order Runge–Kutta numerical integrator [100] to solve Equation 6.1.

6.2.2 Model Parameters

Table 6.2 lists the parameters in the bone-marrow model (the elements in the vector ϕ —Equation 6.2). An important goal is to construct a model in which each parameter represents a real-world physiological concept. The parameter bm_{in} represents the fractional replenishment rate of the bone marrow produced by dividing stem cells (Equation 6.14). Under steady-state conditions and without any drug effect, bm_{in} is equal to the bone-marrow decay rate. When drug effect is present, the effective bone-marrow decay rate is increased by the factor, $d_{sensiv} * drug$. The parameter bm_{in} is the rate of bone marrow replacement (inflow). The parameter bm_{ss} represents the steady-state bone-marrow mass (Equation 6.14). A permanent loss of bone-marrow capacity is modeled as a low bm_{ss} ;

$$(bm_{in} * bm_{ss}) - (bm_{in} * marrow * (1 + (d_{sensitiv} * drug))) \quad (6.14)$$

$$(dline * marrow) - (dline * box1) \quad (6.15)$$

$$(dline * box1) - (dline * box2) \quad (6.16)$$

$$(dline * box2) - (dline * box3) \quad (6.17)$$

$$(wbc_{out} * box3) - (wbc_{out} * wbc) \quad (6.18)$$

$$-1 * d_{out} * drug \quad (6.19)$$

Figure 6.2 State equations in the TOPAZ bone marrow model.

These equations are the elements of the vector $\mathbf{a}$ in Equation 6.1. They encode the temporal relationships represented graphically in Figure 6.1. The first five equations describe the granulocyte kinetics; the last equation describes the drug-effect kinetics. The term $dline$ is equal to $nboxes / (2 + dtransit)$, where $nboxes$ is the number of compartments in the delay chain ($nboxes = 3$ in TOPAZ). Parameters are described in Table 6.2; states are described in Table 6.3. Collectively, these formulae define a non-linear system of first-order differential equations. The predicted peripheral WBC count is the fifth element of the state vector, $\mathbf{z}$.

Table 6.2 Parameters of the TOPAZ bone marrow model.

Parameter name	Parameter units	Initial value	Physiological interpretation
bm_{in}	days^{-1}	0.20	fractional bone-marrow input rate
bm_{ss}	normalized	2.0	bone-marrow steady-state level
$dtransit$	days	2.0	mean delay transit time
wbc_{out}	days^{-1}	0.45	fractional WBC decay rate
d_{out}	days^{-1}	0.075	fractional drug-effect decay rate
$d_{sensitiv}$	unitless	0.25	drug sensitivity slope

A hypercellular marrow is modeled as a high bm_{ss} . The parameter $dtransit$ captures the maturation time for granulocytes (Equations 6.15, 6.16, and 6.17). Granulocytes undergo progressive maturation within the bone marrow before being released into the peripheral circulation. Cells in later stages of development have different sensitivities to myelosuppressive chemotherapy. In the TOPAZ bone-marrow model, only the least developed cells in the bone-marrow compartment are destroyed by myelosuppressive agents. Granulocytes in the delay chain have no additional loss due to the presence of drug effect. The parameter wbc_{out} represents the fractional rate of loss of granulocytes from the circulation (Equation 6.18). The parameter d_{out} encodes the fractional rate of loss of drug effect due to metabolism, excretion, and other processes. Because all four MOPP drugs are combined to form a pseudodrug, this parameter represent the loss of the pseudodrug's myelosuppressive effect and not the loss of any one of the MOPP agents. Finally, the parameter $d_{sensitiv}$ models the patient's sensitivity to the myelosuppressive effects of chemotherapy (Equation 6.19). Two patients receiving an identical dose of chemotherapy and having an identical loss rate of drug effect (d_{out}) may have different responses because of patient-specific differences in their sensitivity to equivalent amounts of drug. The parameter $d_{sensitiv}$ captures this difference as a dose-response relationship between drug effect and drug level. In Equation 6.19, I use a simple linear dose-response relationship. The parameter $d_{sensitiv}$ adjusts the steepness of this response.

Table 6.2 lists the physiological meanings and population-mean values for the parameter vector, ϕ . These estimates were derived from two clinical oncologists and from the medical literature.

Using a value of 2.0 for the parameter bm_{ss} to represent the hypercellular bone

Table 6.3 The TOPAZ bone-marrow model states.

State name	State units	Initial value	Physiological interpretation
<i>marrow</i>	normalized	bm_{ss}	bone-marrow mass
<i>box1</i>	normalized	bm_{ss}	delay line box 1 Mass
<i>box2</i>	normalized	bm_{ss}	delay line box2 mass
<i>box3</i>	normalized	bm_{ss}	delay line box3 mass
<i>wbc</i>	normalized	bm_{ss}	granulocyte mass
<i>drug</i>	normalized	0.0	drug effect

marrow of most Hodgkins' patients is an example of a prototypical model being specialized to a static patient-specific model (Chapter 5). With bm_{ss} equal to 1.0, the TOPAZ bone-marrow model is a prototypical model; with bm_{ss} equal to 2.0, the model is specific to patients with Hodgkins' lymphoma.

6.2.3 Model States

The TOPAZ bone-marrow model presented graphically in Figure 6.1 has six model states: *marrow*, *box1*, *box2*, *box3*, *wbc*, and *drug*. The corresponding state vector $\mathbf{z}$ (Equation 6.1), therefore, has six elements. One state variable in $\mathbf{z}$ corresponds to exactly one of the six model states depicted in Figure 6.1. Table 6.3 lists the six state variables and their initial values ($\mathbf{z}_0$). All model-state variables are normalized. A granulocyte count of 6000 cells per mm^2 converts measured counts into normalized counts. This choice is arbitrary and does not alter the predictions made by the model. Patients with higher-than-average granulocyte counts or larger-than-expected marrow mass have state-variable values greater than 1.0; patients with low peripheral counts or depressed bone-marrow mass have state-variable values less than 1.0.

Because patients with Hodgkins' lymphoma typically have hypercellular marrows, the initial estimated value for the parameter bm_{ss} is equal to 2.0. Therefore, the initial values for all state variables that model marrow quantities are set to 2.0. The initial value for the *drug* state variable is 0. This value implies that the patient has not received any drugs prior to the first visit in the record. If this is not the case, the model accepts a nonzero initial value for the *drug* state variable, representing preexisting drug effect at the

time of the first patient visit.

6.2.4 Model Predictions

Figures 6.3 and 6.4 shows the model predictions ($\mathbf{y}$) using initial states ($\mathbf{z}_0$) from Table 6.3 and the population-based parameter values (ϕ_0) from Table 6.2. Figure 6.3 plots the predicted changes in the bone marrow, WBC, and the three delay-chain compartments, the first five elements, respectively, in the state vector $\mathbf{z}$ (Table 6.3), following a single course of full-dose MOPP chemotherapy. Figure 6.4 plots the predicted changes in only the WBC compartment over three courses of full-dose MOPP chemotherapy. Because the model predictions are generated using population-based parameter values, this second plot depicts the *expected* observed WBC counts of a *typical* patient with Hodgkins' lymphoma who receives multiple courses of MOPP chemotherapy. A patient with an unusual response to treatment will deviate from this expected clinical course. In Chapter 7, I discuss how TOPAZ produces descriptions of unusual clinical events by noting large deviations from this expected clinical course.

6.3 The Model-Fitting Procedures

Prior to obtaining the first granulocyte count, the physician has only population-based information to predict a patient's likely clinical course. The expected clinical course is predicted by the bone-marrow model when population-based parameter and initial state values are supplied (Figures 6.3 and 6.4). Starting with the first measurement, however, the physician begins to accumulate insights about this patient's unique responses to treatment. In this section, I present the techniques that alter the initial population-based model as additional patient observations become available. The key principle is to adjust the model parameter values until the predicted granulocyte values generated by the model with these parameter values more closely match the actual observations than do those predicted by the model with any other set of parameter values. As more data are incorporated into the model, the model parameters change from population-based values to patient-specific values, and the model evolves from a population model to a patient model.

In describing the fitting procedures, I will use the following notation:

1. The vector $\phi_{(T)}$ represents the true parameter values for a specific individual.

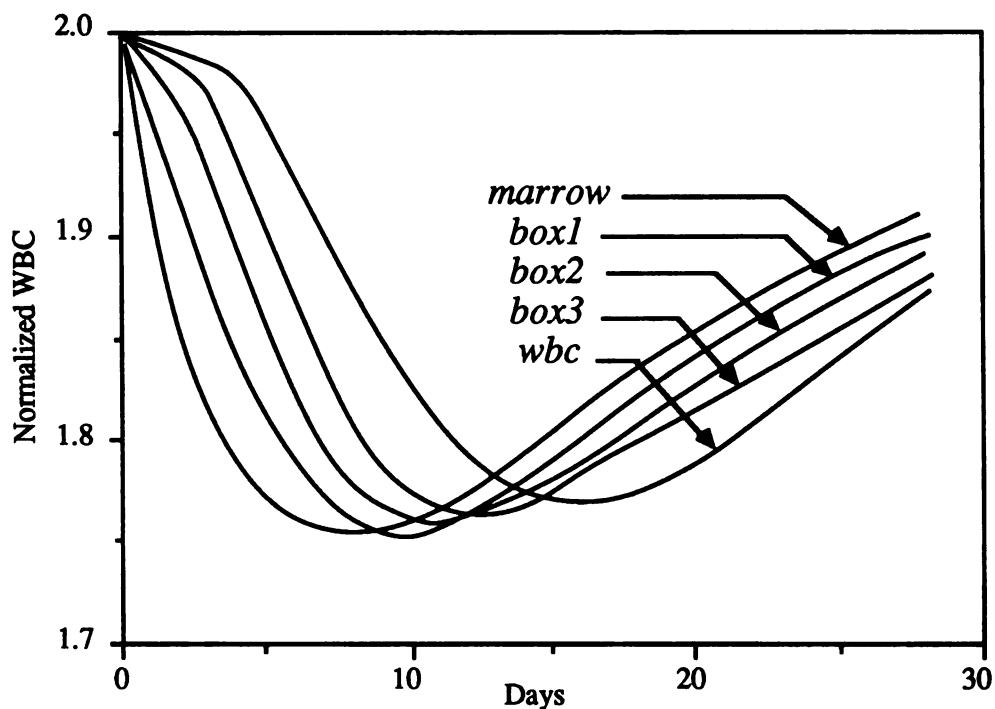


Figure 6.3 Predicted bone-marrow-model states using population parameter values.

This graph plots the predicted changes in five model state variables (*marrow*, *box1*, *box2*, *box3*, and *wbc*) after a single course of full-dose MOPP chemotherapy. Figure 6.4 plots the change in the WBC compartment (*wbc* state variable) over three 28-day MOPP cycles.

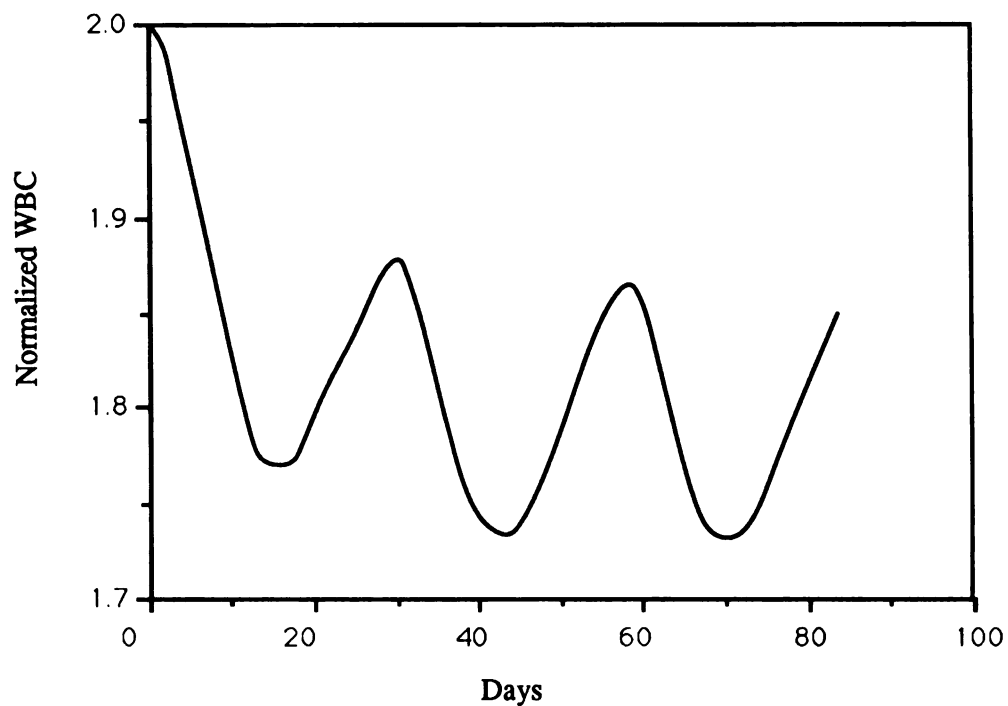


Figure 6.4 Predicted WBC counts following multiple courses of MOPP chemotherapy.

Like Figure 6.3, this figure plots state variable predictions when the TOPAZ bone-marrow-model parameters are set to population values. In this plot, only the predicted WBC counts (*wbc* state variable value) are plotted over the course of three cycles of MOPP chemotherapy. The usual 28-day cycle time between MOPP chemotherapies is used in this plot.

2. Because we cannot directly measure $\phi(T)$, we use model-fitting procedures to estimate a patient's parameter values from the available clinical data. The vector $\hat{\phi}$ represents a vector of estimated parameters. The scalar $\hat{\phi}_i$ represents the i th element in the vector $\hat{\phi}$ or, equivalently, the i th estimated parameter.
3. Because estimated parameter values are derived from observations, a vector of estimated parameters derived from the first k observations is denoted by $\hat{\phi}^{(k)}$. The index k ranges from 1 to the number of available observations. The scalar $\hat{\phi}_i^{(k)}$ represents the i th element in the vector $\hat{\phi}^{(k)}$ or, equivalently, the i th estimated parameter derived from the first k observations.
4. Model-generated predicted WBC values are denoted as $\hat{y}$. From Equation 6.1, $\hat{y} = \mathbf{z}'\mathbf{C}\psi$. For a vector of predicted WBC values, this equation becomes $\hat{\mathbf{y}} = \mathbf{Z}'\mathbf{C}\psi$. Because $\mathbf{z}$ and ψ are functions of a specific parameter vector, ϕ , $\hat{\mathbf{y}}^{(\phi)}$ denotes the model predictions calculated using the parameters ϕ . An equivalent notation for model predictions generated from estimated parameters $\hat{\phi}$ is $\mathbf{y}(\hat{\phi})$.
5. The vector $\mathbf{y}$ is equal to $(y_{(1)}, y_{(2)}, \dots, y_{(n)})$, where $y_{(i)}$ is a patient observation at t_i . With this definition for the vector $\mathbf{y}$, $y_{(i)} = y_i$.

In TOPAZ, the model-fitting procedures derive patient-specific estimates for model parameters ($\hat{\phi}$) by comparing observed WBC counts (y_i) to model-predicted WBC counts (the fifth element in the state vector $\mathbf{z}_{(i)} = \hat{y}_{(i)}$). Because we directly observe one of the system states, from Equation 6.3, the design matrix $\mathbf{C}$ contains all zeros, except for a 1 in the fifth row of the first column ($c_{5,1} = 1$).

6.3.1 Least-Squares Estimation

In TOPAZ, I use weighted least-squares (WLS) to derive estimated model parameter values ($\hat{\phi}$) based on patient observations. Least-squares-estimation procedures determine the set of parameter values that minimize a least-squares objective function. The standard WLS objective function to be minimized is

$$\mathcal{O}_{WLS}(\phi) = \sum_{i=1}^n w_i [y_i - \hat{y}_i]^2 \quad (6.20)$$

Using the pseudo-data approach to a Bayesian formulation, the standard WLS objective function is modified to include a second penalty term that adds additional weight for differences between the current parameter estimates and the prior mean (ϕ_0) [107] :

$$\mathcal{O}_{WLS}(\phi) = \sum_{i=1}^n w_i [y_i - \hat{y}_i]^2 + \sum_{j=1}^m u_j [\phi_{0,j} - \phi_j]^2 \quad (6.21)$$

In Equation 6.21, n is the number of observations, m is the number of model parameters, w are weights assigned to the observations, and u are weights assigned to parameters. The estimated parameters, $\hat{\phi}$, are the set of parameters that minimize Equation 6.21. The Levenberg–Marquardt method solves the minimization problem [100].¹

Because the calculation of weights alters the final parameter estimates, this issue has received considerable attention in statistics. I use the most common form of weighting, called *reciprocal variance weighting*, which states that

$$w_i^{-1} = \text{var}(y_i) = \text{var}(\varepsilon_i) = \sigma^2 * v(z_i) \quad (6.22)$$

The term $v(z)$ is the variance submodel. Reciprocal variance weighting is intuitively attractive because it gives higher weights to those terms with lower variance. Thus, reciprocal variance weighting gives more influence to terms in which we have more confidence. From Equation 6.4, $v(z) = (z' C \psi)^2$. To estimate z for the purpose of estimating $v(z)$, I use the population-mean parameter values, ϕ_0 . From Equation 6.22, the weights used for observations in Equation 6.21 are

$$w_i^{-1} = (0.06 * z_i' C \psi_i)^2 = (0.06 * \hat{y}_i(\phi_0))^2 \quad (6.23)$$

For parameter terms, from Equation 6.12

$$u_j^{-1} = \text{var}(\phi_{(T)j}) = \text{var}(\eta_j) = (0.3 * \phi_{0,j})^2 \quad (6.24)$$

Given the previous equations, a collection of n observations (y), and p independent variables (X), Equation 6.21 can be minimized with respect to ϕ to derive new model parameter estimates ($\hat{\phi}(n)$) that incorporate both the prior parameter estimates and the patient-specific observations. The complete fitting algorithm used in TOPAZ is as follows:

¹Lewis B. Sheiner supplied the initial FORTRAN routines that implement the Bayesian weighted least-squares algorithm.

1. Do an initial fit for robust estimation procedures
 - a. Turn off flags for robust estimation techniques
 - b. Calculate initial weights, w_{init} and u_{init} , according to Equations 6.23 and 6.24
 - c. Do WLS fit with these initial weights, yielding initial estimated parameters, $\hat{\phi}_{init}$
 - d. With initial estimated parameters, $\hat{\phi}_{init}$, calculate initial model predictions, $y(\hat{\phi}_{init})$
 - e. Turn on flags for robust estimation techniques
2. Calculate final weights, w and u , using initial weights, w_{init} , u_{init} , $\hat{\phi}_{init}$, $y(\hat{\phi}_{init})$, and flagged robust estimation techniques
3. Do WLS fit yielding estimated parameters, $\hat{\phi}$
4. Calculate covariances

I discuss the justification for steps 1a through 1e in Section 6.4.2. In the next section, I present the robust estimation techniques used in step 2. With step 1 removed, or (equivalently) with no robust estimation flags set, the algorithm is a simple WLS-fitting procedure.

Certain physiological parameters in the bone-marrow model represent physical processes that have only positive values (bm_{in} , wbc_{out} , and d_{out} —Table 6.2). To ensure positive-valued estimates, a procedure log transforms these three parameters before the fit and then untransforms them after the fit. Since logs can be positive or negative, but antilogs are always positive, this technique ensures that the physical constraints are enforced.

6.3.2 Robust Estimation

The calculation of the weights, w and u , in Equation 6.21 is a critical step in obtaining the best possible fit from a set of data. The relative size of weights for observations (elements in the vector w) compared to weights for parameters (elements in the vector u) will change the model's responsiveness to new observations. Note that the presence of one observation with a large weight relative to all the other observations forces the fitting procedures to find a set of parameters that closely fits the high-weight observation at the expense of a poorer fit to other observations. The size of the weights, therefore,

determines how much influence an observation or a prior parameter estimate has on the set of parameters that minimizes Equation 6.21.

Robust estimation techniques modify the fitting procedure to be less sensitive to erroneous or outlying observations (step 2). Because the objective function penalizes predictions based on the squared distance between observations and model predictions, an outlying observation can have an inordinate influence on the final set of fitted model parameters. Robust estimation techniques usually modify the weights given to observations that appear to be highly unexpected. The general approach is to inspect the size of the scaled residual of the suspect observation. If the scaled residual is much larger than is expected given the expected magnitude of ε (the error term in Equation 6.3), then the weight assigned to this observation is decreased. I used the robust estimation technique in TOPAZ.

For the n th fit, a set of initial weights, w_{init} and u_{init} , and estimated model parameters, $\hat{\phi}_{init}$, using all observations ($y_i, i = 1, n$), is obtained (steps 1a through 1e). These parameters are used to calculate initial predicted values ($y(\hat{\phi}_{init})$). In calculating the weight used for the i th observation in step 2, I calculated the residual for this observation using the initial weights and the initial predicted values:

$$r_i = | (y_i - \hat{y}_i(\hat{\phi}_{init})) | * \sqrt{w_{init\ i}} \quad (6.25)$$

A large value for the residual, r_i , implies that, given all n observations, the i th observation is very unexpected. If r_i exceeds a predefined threshold, an attenuation factor that decreases from 1 to 0 reduces $w_{init\ i}$ to yield a smaller weight, w_i . In TOPAZ, I use a threshold of 2.0. From Equation 6.25, this threshold value means that an observation with a residual larger than two standard deviations from its initial prediction ($y(\hat{\phi}_{init})_i$) is penalized with a smaller weight. The attenuation factor ($\text{threshold}^2 / r_i^2$) applies progressively more down-weighting the further an observation is from two standard deviations past the model prediction. The effect of this down-weighting scheme is (approximately) to prevent any observation from contributing to the final objective function more than would an observation with $r_i = 2.0$.

A second robust estimation procedure protects against excessive weights being assigned to small-valued observations. Using the constant-CV variance model, the weight for the i th observation is (Equation 6.23)

$$w_i^{-1} = (0.06 * \hat{y}_i(\phi_0))^2$$

As $\hat{y}_i$ becomes small, the constant-CV variance model probably becomes incorrect as it overestimates the weight for small-valued predictions. I bound the maximum weight that can be assigned to any single observation. The upper bound is a fraction of the weight for the largest observation. Because patients receiving chemotherapy tend to have the highest granulocyte count prior to treatment, I simplify this calculation by using a fraction of the first observation (y_1) to determine the upper weight bound. Thus, Equation 6.23 is modified to

$$w_i^{-1} = \left(\max \left[0.06 * \hat{y}(\phi_0); (\xi * y_1) \right] \right)^2 \quad (6.26)$$

The implication of this modified weighting scheme is that the highest weight that can be assigned to any observation is the same as the weight assigned to an observation equal to $(\xi * y_1)$. In TOPAZ, $\xi = 0.333$.

6.3.3 Examples of Model Fitting

Before patient data are used to alter model parameters, only population-based parameter values are available to generate model predictions. Using population parameters and the timing and dosage of MOPP chemotherapy in a patient's medical record, the TOPAZ bone-marrow model simulates the *expected* changes in granulocyte counts due to chemotherapy for a patient with population-based parameter values. Because the population parameter values represent the *typical* patient, the changes in granulocyte counts predicted with population parameters model the *expected* clinical course. New patient data alter the model parameters to predict the actual observations more accurately. Figure 6.5 shows a series of model predictions with different model parameter values. In each plot, the actual granulocyte counts are marked and the predicted values are plotted on the solid line. Figure 6.5(a) graphs the model predictions using the population parameter values; plots 6.5(b) through (e) graph model predictions using parameter values calculated by the fitting procedures described in Section 6.3.1. The patient data used to generate these parameter estimates were taken from the medical record of the patient summarized in Figure 5.7. Note that better model predictions occur with estimated parameter values derived from

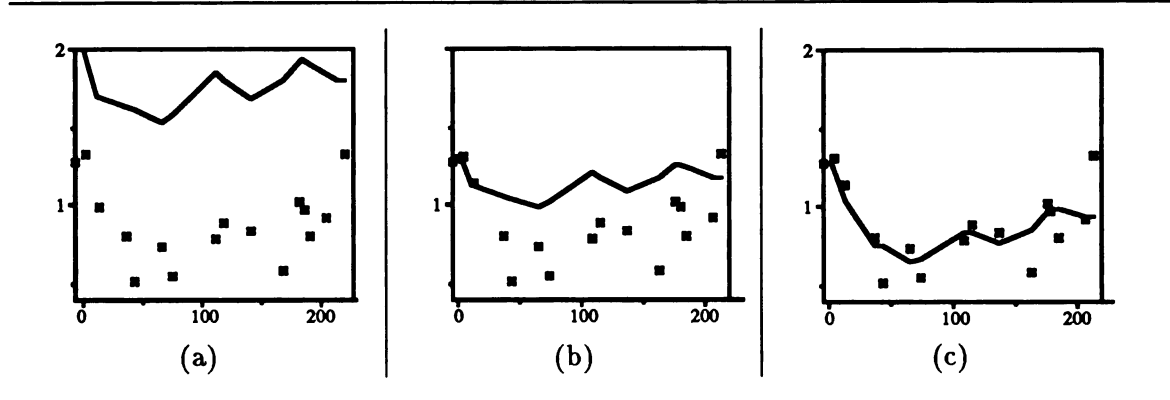


Figure 6.5 Predicted white blood cell counts using patient parameters.

Observed WBC counts are marked; predicted counts are connected by a solid line. Plot (a) uses population-based values for all parameters; plots (b) and (c) incorporate successively more patient observations.

fitting of more patient observations. Figure 6.5 depicts the evolution of a population-based model of granulocyte changes with MOPP chemotherapy to a patient-specific model as more patient data alter the initial parameter estimates.

6.4 Problems and Extensions

The previous sections examined the features of the TOPAZ bone-marrow model (the structural model) and the fitting procedures that estimate patient-specific parameter values. In practice, this separation is not distinct. The fitting procedures are able to alter only those parameters that are represented in the structural model. The fitting procedures also contain assumptions about the likely distributions of errors in measurements and parameters (the variance models) in order to define and find optimal estimated parameter values. Failure to fit the model predictions closely to the observed data can arise from problems in the relationships represented in the structural model, in the distributional assumptions in the variance models, or in both. Identifying the party responsible for a poor fit is often impossible.

The structural model and the fitting procedures are critical to the abstraction task that follows (Chapter 7). The bone-marrow model determines which features in the real world can be examined and abstracted. The fit determines the confidence that can be

placed on statements based on estimated model parameters and on model predictions. In the next section, I examine inadequacies in the TOPAZ physiological model and propose areas of further work that address some of these issues. Problems and solutions in both the structural and variance model are examined.

6.4.1 Model Inadequacies

A structural model is a limited representation of reality. The less complex the model, the more situations in which it does not properly represent the key entities and relationships in the real world. It is important that the structural model represent the most important features accurately and adequately. Because models are always incomplete, it is also critical to have techniques that detect situations in which the model performance is likely to be poor.

Most measures of model performance examine the ability of the model predictions to match the observations. The better the match, on average and over a number of test cases, the better the model. We can improve any model by increasing its complexity to include rare or unusual situations. The added complexity has a price, usually in increased computing time or increased model instability. For example, I constructed a structural model that was similar to the model shown in Figure 6.2 but that contained seven parameters rather than six (Figure 6.6). This new model has one additional compartment for modeling cumulative toxicities following repeating cycles of MOPP chemotherapy. With this simple addition to the model, the parameter estimates become unstable, changing widely with each new patient observation. This inability to determine stable parameter estimates is a common problem with overparameterized models, where there are many possible parameter solutions that are nearly equally acceptable to the fitting routine. Each new datum favors one solution, and then another, and then a third. The problem arises when the data do not have sufficient information to discriminate among the subtle differences that can be represented in a complex model. How does the model builder decide how much complexity is enough to capture the necessary concepts, but is not excessive for the data that are available for his problem? There is no general solution to this problem.

The TOPAZ bone-marrow model encodes only a tiny fraction of the existing medical knowledge about cancer chemotherapy and granulocyte kinetics. Key features not represented in the current model include

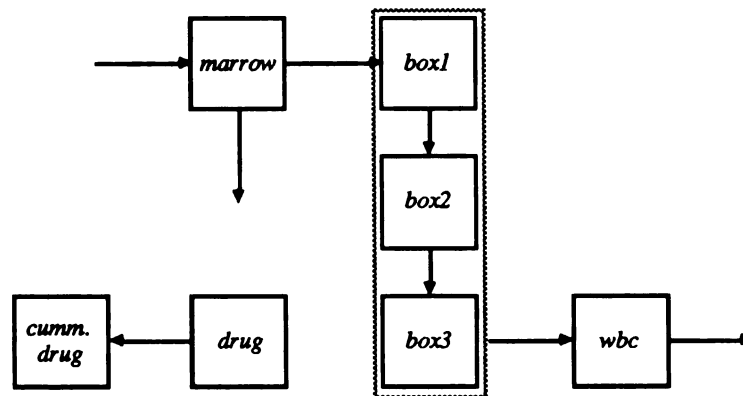


Figure 6.6 A seven-compartment bone-marrow model.

This model is an elaboration of the model in Figure 6.1. A seventh compartment for cumulative drug dose has been added. This addition enlarges the model by one equation and one parameter. The result is an unstable, overparameterized model that results in erratic parameter estimates.

- Cumulative chronic bone-marrow toxicity (see Figure 6.6)
- Maturation-chain cell destruction
- Metabolism or dynamics of individual chemotherapeutic agents
- Distinction between drug-mass and drug-effect models
- Nonchemotherapy causes of WBC changes (infection, drugs, bleeding)

Items in this list are real physiological processes that have no representation in the current structural model and therefore have no representation in a TOPAZ summary. In rare but possible clinical situations, these missing entities may be the primary cause for the observed counts. The fitting routines fit the model by adjusting only those physiological parameters that are explicitly represented in the current bone-marrow model. If the offending physiological process has no parameter in the current model, the other parameters must change to match the observations. The fitting procedures may find a set of parameter values that successfully minimizes the WLS objective function (Equation 6.21) but misrepresents the true state of the patient's underlying physiology.

6.4.2 Parameter Assumptions

The TOPAZ bone-marrow model assumes stationary parameter values ($\Gamma = 0$). A stationary model assumes nonchanging true parameter values ($\phi_{(T)}$)—only our best estimate of their value ($\hat{\phi}$) varies with additional observations. Because we become more informed about the parameter's true value with more observations, a plot of the estimated parameter value versus the number of observations ($\hat{\phi}^{(n)}$ versus n) should approach the true value asymptotically (Figure 6.7). Patients almost certainly violate the assumption of stationary parameter values. In Figure 6.7, I plot a series of actual parameter estimates from a patient who had a difficult clinical course late in his treatment. Note how the parameter estimates initially appear to be approaching a maximum value, but then fall substantially by the end of the record. Early in the clinical course, the data suggest a high parameter value. Successive observations support a high parameter value. Later in the course, the data contradict the high estimated parameter value, so the estimate begins to drop. There are two interpretations for a change in the direction of a parameter estimate as the number of observations increases: (1) there is a systematic bias in the initial data, not due to the model parameter, that makes the higher parameter estimate plausible; and (2) the parameter being estimated changes value during the clinical course, resulting in data that no longer support the initial parameter estimate. In medicine, the latter explanation almost always is correct. Mathematical techniques for modeling time-changing parameters exist, such as a state-space model with process noise. In these models, Equation 6.13 is no longer valid. In general, estimation techniques for structural models with process noise (for example, the nonlinear Kalman filter) are both data-intensive and computationally expensive. The patient records I use have nine to 16 observations taken over 6 months to 2 years. The TOPAZ problem is data-poor and cannot support the data requirements of more sophisticated modeling methods. Data-rich medical environments, such as the hemodynamic monitoring of patients in the intensive-care unit, may benefit from the use of other models that do not embody the assumption of stationary parameter values.

The erroneous assumption of stationary parameters results in parameter estimates that change slowly even if there is strong evidence to support a different parameter value. The current fitting algorithm uses all available patient data to estimate parameter values. If some of the data support one parameter value and other data support a different value, the current fitting procedures settle on an estimated value somewhere between the two ex-

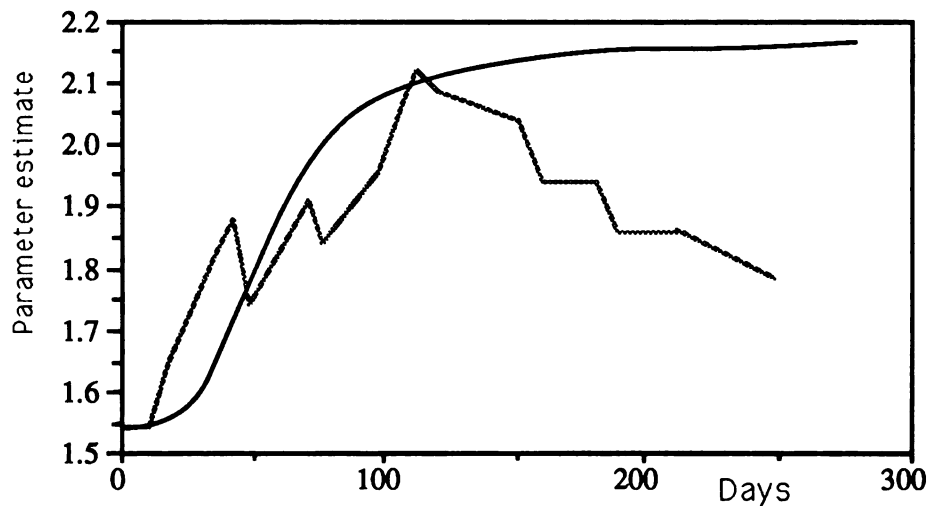


Figure 6.7 Changes in a parameter estimate with increasing observations.

This figure graphs the estimated value of a parameter as additional patient observations are included in the fit. The solid-line plot, derived from simulated data, smoothly approaches an asymptotic value, indicating that additional information supports and refines the current parameter estimate. The dashed-line plot, derived from patient data, moves toward a similar value but then changes direction, indicating additional information does not support the initial high parameter estimate. In this case, the patient's true parameter value probably changed during the course of treatment. Unlike most examples in this chapter, this graph was not constructed from the patient record summarized in Figure 5.7.

tremes. The ideal solution in this situation is to construct a more complete model that permits time-varying parameters; in our data-poor setting, however, a series of changes to the model-fitting procedures were investigated to simulate the desired behavior of a true process-noise model. The key issue is to find an alternative method that gives the proper weights, and therefore the proper influence on the fitting procedures, to observations that initially contradict the current parameter estimate, but in retrospect, after confirmatory observations, indicate a true change in the patient's parameters. We seek a method that initially ignores contradictory observations until there is sufficient evidence to suggest that we instead should start ignoring the previous observations. Two ad hoc alterations to the fitting procedures, designed to increase the model's responsiveness to real changes in parameter values while limiting responsiveness to truly erroneous observations, were implemented and tested. In TOPAZ, both methods resulted in unstable parameter fits, and so were not used in the final fitting calculations.

If past observations have too much influence on the current parameter estimates, one solution is to eliminate old data from the fit. The fit uses only the last m observations rather than all n observations. This method is particularly attractive because the fitting procedures use at most m observations to find a solution even if there are many more observations available. The size of the fitting problem, which usually grows as the number of observations to be fitted increases, remains fixed with this technique. Like the standard fitting procedure, the altered fitting procedure uses the initial population parameter values (ϕ_0) and the initial population covariance matrix (Ω), but the new method fits only observations from y_{n-m} to y_n rather than those from y_1 to y_n . This technique was tried with $m = 1, 2$, and 3 . Unfortunately, in all cases, the fits were too sensitive to the current observations, and parameter estimates varied widely with each new observation. A more continuous solution to decreasing the influence of past observations on current parameter estimates is to lower the weights of old observations using a time-decay function. This scheme assumes that clinical data collected in the distant past have less relevance to current parameter values than do data collected more recently. A time-decay function applies greater weight attenuation the further in the past an observation is from the current time. A simple exponential decay function, $e^{-k*(t-t_i)}$, calculates this attenuation factor. The parameter k determines the rate at which past values are down-weighted. In this method, all data are used in the fit, but past data have attenuated influence. This model has the unfortunate but explainable property that, as the time since the last measurement increases, the

down-weighting of past data causes the parameter estimates to return to the initial prior parameter values. In TOPAZ, following a long delay between clinic visits, the parameter estimates drift back toward the population-based priors (ϕ_0). This return to the population priors occurs even if the most recent parameter estimates were substantially different from the population values. The reason for this drift is that, with time-varying parameters and without new information to the contrary, our knowledge about the parameter values returns toward the state of prior information. The rate of drift back to ϕ_0 is controlled by the characteristics of the time-decay function. A long time-decay slows the drift, but it also diminishes the down-weighting of less relevant past observations.

Making the model more responsive to current observations also makes the model more vulnerable to the influence of spurious or outlier observations. Robust estimation techniques attempt to deal with markedly unexpected observations, which may unduly bias the fit. This problem is the opposite of that of time-varying parameter values. With robust estimation techniques, we wish to *prevent* the fit from responding to an observation. I implemented two ad hoc methods to address this problem. One method produces acceptable fits and is used in TOPAZ; the other was abandoned.

Using the same logic for down-weighting past observations, it is possible to down-weight only the j th observation if that observation's value seems unusual. I calculate the residual for the j th observation using the parameter estimates calculated using only the previous $j - 1$ observations. Using a formula similar to Equation 6.25, the weight for the j th observation is decreased if there is a large residual. Experience with this down-weighting scheme reveals that it successfully smooths the parameter estimates even when the data have wide swings. Unfortunately, an inextricable property of this robust estimation technique is that the model estimates also ignore real parameter changes present in the data. An observation that signals a real parameter change is as unexpected as is a truly erroneous measurement. Down-weighting minimizes the effect of this observation. When a second confirmatory measurement appears, the model considers this value to be erroneous and also incorrectly down-weights the value's influence on the fit. Three or four consecutive measurements are required before the model fit reacts to the change. In the data-poor TOPAZ environment, this method is too conservative to be useful.

The goal is to develop a method that initially assumes that an unexpected observation is probably an erroneous or unusual measurement, but revises this assumption if subsequent observations confirm the change. The two-fit algorithm described in Section 6

(page 154) is designed to emulate this behavior. The initial fit in step 1 uses all observations, including the possibly unexpected j th measurement, to calculate an initial set of estimated parameter values ($\hat{\phi}_{init}$). All robust estimation techniques are turned off, so each observation contributes its full weight to this fit. This initial parameter set is used to calculate the residuals in step 2. Given disabling of the robust estimation procedures, a series of unexpected but confirmatory observations can alter the parameter values in $\hat{\phi}_{init}$ calculated in the initial fit, so that previously unexpected observations begin to be predicted (and expected) by the model. The residuals for these observations become smaller as additional confirmatory observations continue to change the model predictions toward these values. With smaller residuals, the robust estimation procedures in step 2 do not attenuate these observations. Earlier observations that initially were acceptable may become unusual in light of later observations and will be down-weighted. In this way, observations that early in the clinical course seemed likely but now appear unlikely are down-weighted, and observations that earlier seemed unusual but now appear likely maintain a normal weight.

A highly desired modification to the fitting procedures would be an algorithm that gives acceptable parameter estimates and has a fixed computational complexity as the number of data increases. The current fitting procedures become computationally more difficult as the number of observations to be fitted increases. Although many optimizations to the current FORTRAN code could reduce the current computation time, the complexity continues to increase with more observations. Limiting the fit to a fixed number of measurements results in erratic parameter estimates that are strongly influenced by the most recent observation. In a data-intensive setting, some method must be used to either summarize past observations or to limit the number of past observations needed to derive stable estimates. In the data-poor TOPAZ environment, the various approximation techniques that I tried lost too much information to be useful.

CHAPTER 7

An Interval Model of Temporal Abstraction

An understanding of surprising observations is a critical component of effective therapeutic decision making. The mathematical techniques presented in Chapter 6 provide an intuitive method for detecting unusual observations. Model predictions with population parameters (ϕ_0) are the *expected* WBC counts for the *typical* patient. Measured WBC counts and patient-specific model predictions that are significantly different from the expected WBC counts are *unusual*. The structural model ($\mathbf{z}'\mathbf{C}\psi$) and the patient-specific parameters ($\hat{\phi}$) provide a mechanism for explaining why the unusual observations occurred based on the underlying physiological processes contained in the model. In this section, I present a second modeling technique, based on symbolic temporal pattern matching, designed to derive the abstract clinical concepts implied by unexpected patient observations, abnormal model parameters, and unusual model predictions.

Abstracting numbers into meaningful concepts requires a model that has characteristics that are markedly different from those of the physiological bone-marrow model. The goal of this second model is to derive aggregates of time during which surprising events were seen. These key events form the basis of a summary of the patient's clinical course. TNET is a representation technique used in ONCOCIN to encode the presence of

therapeutic events during a patient’s clinical course (Chapter 4). In TOPAZ, the intervals represented in TNET are generalized to include nontherapeutic events inferred from patient observations using the patient-specific model presented in Chapter 6. For example, periods of severe myelosuppression or excessive drug effect are inferred events that are represented in TOPAZ but are not represented in ONCOCIN. In this discussion, the term *context* is used to denote the general notion of a well-defined interval of time that has clinical meaning. A *context model* encodes the characteristics that define the existence of a context and the important features that describe each instance of a context. The context model in TOPAZ manipulates numbers, such as patient observations and model predictions, and concurrent or dependent contexts in a patient’s record to construct clinical abstractions. ETNET, a reimplement and major extension of TNET, encodes context models in TOPAZ.

7.1 Interval-Based Context Models

Process-based temporal models, such as the TOPAZ bone-marrow model (Chapter 6), encode time as a continuous quantity. Given a set of initial conditions, parameter values, new inputs, and constraints among variables, the bone-marrow model produces predicted model states at any time t . Missing from this perspective of time is the *meaning* of the predicted states or an *understanding* of intervening predicted states. A very low predicted or observed WBC count has clinical importance that has no representation in a process-based model. A drug sensitivity that is estimated to be higher than is usual alters the TOPAZ bone-marrow model predictions, but nowhere does this model encode the clinical knowledge that a significant change in this estimated parameter value has important therapeutic ramifications. Process models do not represent the *implications* of their predictions; context models encode the domain-specific knowledge required to *interpret* the observations and process-model predictions.

In dynamic domains, observations and model predictions change over time. The implications of an observation or of a model prediction also change over time. A context model provides definitions for events that have similar characteristics and are worth noting. Domain-specific knowledge defines which findings constitute *similar* characteristics and which intervals are *worth* noting. It is the role of the context model to encode this knowledge in a way that the existence and termination of these interesting dynamic events can be detected.

Defining a context model is more complicated than is merely enumerating a set of observed or predicted numbers that are worth noting. In one context, a set of numbers may imply one event, whereas in a different context, the same set of numbers may imply another event. For example, we may observe a patient who has very low WBC counts after chemotherapy. Can we infer that this patient is very sensitive to the myelosuppressive effects of treatment based on these observations? If the patient received twice the normal dose of drugs, then, in this context, we cannot make this inference. If the patient received one-half of the normal dose of drugs, then, in this other context, we have strong evidence for the inference. The bone-marrow model described in Chapter 6 would not alter the patient's estimated drug-sensitivity parameter in the first case and would alter this parameter in the second case. An abstraction system that did not contain context-sensitive knowledge would not be able to distinguish between these two markedly different interpretations. Therefore, in addition to defining which features comprise an interesting event, a context model must contain knowledge about the contexts in which this definition is correct.

7.2 ETNET

The TOPAZ context model has two main tasks: (1) to abstract clinically meaningful intervals of time from the medical record, and (2) to explain these events using information supplied by the patient-specific bone-marrow model. The TOPAZ context model is encoded in a structure called ETNET. ETNET is derived from and an extension of TNET (Chapter 4). TNET is a temporal structure that helps ONCOCIN to perform complex temporal queries for time-ordered, context-sensitive data. The presence of TNET intervals, however, does not alter the reasoning performed by the ONCOCIN REASONER. ETNET extends the capabilities of TNET to include context-specific rules to conclude key features about an interval (such as maximum or minimum values within the interval) and rules that determine if other context-sensitive intervals can be created. Unlike TNET, ETNET is an active reasoning structure. In TOPAZ, ETNET models a patient's hematologic changes as a set of intervals. Each interval represents the existence of a notable clinical event or physiological state. Before presenting the TOPAZ context model, I shall describe the technical details of ETNET.

An ETNET temporal network is a collection of OZONE objects, called ETNODEs. The basic structure of an ETNODE is less complex than that of a TNODE (cf. Figure 7.1

```

(TNODE
  (START      83.11.30)
  (STOP       84.05.17)
  (TYPE       Residuals)
  (SUBTYPE    Obs.v.Pop-))
(TRESULT
  (DURATION   169)
  (MAX        84.12.06))

```

Figure 7.1 The structure of a basic ETNODE.

An ETNODE is an OZONE object with two new spaces: TNODE and TRESULT. ETNODEs are less complex than are TNODEs. Compare this figure to Figure 4.6.

versus Figure 4.6). An ETNODE contains two OZONE spaces, TNODE and TRESULT. The TNODE space has four slots—START, STOP, TYPE, and SUBTYPE—which contain the same information as TNET objects. For the START and STOP slot values, a new dating scheme, adapted from Ladkin [66], is used. a date is encoded as a simple YEAR.MONTH.DAY label. This data representation is human readable and still maintains the proper chronology of dates using simple lexicographic ordering. One or both dates can be a time in the past, present, or future. Two distinguished dates, *past* and *future*, represent times that extend beyond all other dates. In practice, a new interval frequently is created with a STOP time initially set to *future* until some subsequent event, later in the clinical course, terminates the interval. In ETNET, rules terminate ETNODEs by resetting the STOP field to a new termination or stop date.

Unlike ONCOCIN TNODEs, ETNODEs do not have doubly linked lists that connect temporal sequences of nodes. In addition, ETNET does not explicitly represent contextual-dependency hierarchies. This design change from TNET removed a number of restrictions present in the earlier design (Section 4.5). For example, TNET permitted new intervals to be added only to the *leading edge* of the network. Because there are no complex lists to maintain, new intervals can be added anywhere in ETNET. The TOPAZ context

model uses this new feature extensively. In TOPAZ, the entire medical record is used to search for defined temporal patterns. Rules can specify searches to occur from earlier dates to later dates or from later dates to earlier date (Section 7.2.3). TOPAZ exploits the freedom to return to a previous portion of the record to confirm or deny a potential event. Intervals may be created anywhere in the network.

Although ETNODEs do not contain the lists found in TNODEs, there still is a need for finding sibling ETNODEs that precede or follow an ETNODE and for searching temporally dependent children ETNODEs. To support the added simplicity of ETNET and still to maintain the features of the ONCOCIN TNET, each ETNET has a unique node, called a TOPNODE. A TOPNODE is not an ETNODE because its structure is markedly different and it does not represent an interval of time. A TOPNODE contains multiple lists of ETNODEs categorized by date and by ETNODE type labels.

7.2.1 Context-Specific Reasoning

Without any further additions, ETNET would merely be a more flexible reimplementation of TNET. The key new feature that I add in ETNET is the ability to associate reasoning methods with ETNET intervals. Placing reasoning methods within intervals gives ETNET a new active dimension that is not present in TNET.

An ETNODE represents an interval of time during which a particular context or event is present. A context defines features that are of particular interest when that context exists, or triggers a search for closely associated events. For example, during a period of lower-than-expected WBC counts, the lowest recorded WBC value and the duration of the low counts are important features of that context. In addition, a search for model features that could explain the cause of the depressed counts should be initiated. Without the presence of the triggering context, it would not be reasonable to spend time looking for related features or events. The context represented by an ETNODE determines which features of the patient data or of the patient-specific model are worth looking at.

ETNET implements context-specific reasoning by associating rules in ETNODEs. ETNODEs with rules have an additional OZONE space called TOPAZ (Figure 7.2), which contains two new slots, DATERULE and FEATURERULE. Both slots contain a list of RULENODEs that execute within the temporal extent of that ETNODE. The detailed structure of a RULENODE is described in Section 7.2.2. DATERULEs are RULENODEs that

contain predicates that, if satisfied, trigger the creation of other intervals. The term *parent* ETNODE refers to the interval containing the list of DATERULEs; the term *child* ETNODE refers to the interval created by the successful firing of a parent ETNODE DATERULE. Because DATERULE predicates are executed only over the temporal extent of the parent ETNODE, children intervals created by successful DATERULEs are contextually dependent on the parent interval. Each child ETNODE has its own DATERULEs list which may, in turn, create other children intervals. Unlike DATERULEs, FEATURERULEs are RULENODEs that do not instantiate new intervals. If the predicates in a FEATURERULE are satisfied, some feature about the ETNODE interval is asserted. For example, when an interval representing the presence of falling granulocyte count is created, FEATURERULEs may determine whether the fall is large or small, rapid or slow, significant or insignificant. The actual definitions of these descriptive terms are context-specific; FEATURERULEs associate the definitions of these terms with the context.

7.2.2 ETNET RULENODES

RULENODEs are OZONE objects that implement context-specific reasoning in ETNET. RULENODEs do not represent intervals of time, so they are not ETNODEs. A RULENODE is associated with an ETNODE by being named in a DATERULE or FEATURERULE list (Figure 7.2). Multiple ETNODEs can contain the same RULENODE. In this setting, each instance of the RULENODE is evaluated independently from instances in other ETNODEs.

RULENODEs are OZONE objects with three object spaces: TOPAZ, PREDICATES, and ACTIONS (Figure 7.3). The TOPAZ space has information that specifies how the rule predicates should be applied within the temporal extent of the ETNODE that contains the RULENODE. The PREDICATES slot specifies the conditions that must exist for the ACTIONS slot to be executed.

The TOPAZ space in RULENODEs has three slots: CONTEXT-TYPE, RULE-TYPE, and TIMEDIRECTION. The CONTEXT-TYPE slot contains an ETNODE type label. When a new ETNODE is created, RULENODEs that have the same type label in their CONTEXT-TYPE field as the new ETNODE are added to that ETNODE's list of rules. RULE-TYPE contains a single value—either the string *FeatureRule* or the string *DateRule*—which determines on what list in the ETNODE this RULENODE is to be placed.

```

(TNODE
  (START      83.12.06)
  (STOP       84.05.24)
  (TYPE       DElevel)
  (SUBTYPE    Elevated))
(TRESULT
  (DURATION   170)
  (MAX        84.12.06))
(TOPAZ
  (DATERULE   (AbnormalPDrug AbnormalSensitivity
               Summarize.DE.residuals))
  (FEATURERULE
    (DE-level-increase-large DE-level-increase-rapid
      Drug.residual.runs)))

```

Figure 7.2 An ETNODE with context-specific rules.

An ETNODE with context-specific reasoning methods contains an additional space, called TOPAZ, that has two new slots, DATERULE and FEATURERULE. These slots contain lists of RULENODEs that may be executed anywhere within the temporal extent of ETNODE.

NAME: AbnormalPdrug

(TOPAZ

(RULETYPE DateRule)

(TIMEDIRECTION LL)

(CONTEXT-TYPE SummarizePatientRecord))

(PREDICATES

(P1 (INRANGE? MVALUE PDRUG TODAY 2.0 100.0))

(P2 (NOTEXISTS? AbnromalParameter DrugEffect
 TODAY)))

(ACTIONS

(A1 (NEWINTERVAL AbnormalParameter DrugEffect))

(A2 (ASSERT.MIN.VALUE P3)))

Figure 7.3 An ETNET RULENODE.

RULENODEs have three specialized OZONE spaces: TOPAZ, PREDICATES, and ACTIONS. The TOPAZ space contains information that specifies which ETNODEs can contain this rule and which temporal-searching strategy should be used to match predicates. The syntax for the PREDICATES and ACTIONS is detailed in Appendix B.

The TIMEDIRECTION slot determines the direction of temporal searching. The contents and behavior of this slot are discussed in Section 7.2.3.

The PREDICATES slot contains one or more executable LISP expressions. All PREDICATES slots must succeed for the rule to “fire.” A temporal-searching algorithm controls how the pattern matching is performed (Section 7.2.3). The ACTIONS slot contains one or more executable LISP expressions that are executed sequentially if the rule succeeds. No direct chaining of RULENODEs occurs in ETNET. However, the creation of an interval can lead to the addition of new rules that cause the creation of other intervals in a depth-first, data-driven manner. A listing of rule predicates and actions is provided in Appendix B.

7.2.3 Temporal Pattern-Matching and Searching

A recursive algorithm, called INSTANTIATE-ETNET-INTERVALS (IEI), is executed by each newly created ETNODE (Figure 7.4). This algorithm creates the actual abstraction intervals. The IEI algorithm creates a new interval, asserts the rules for that interval's context into the interval, and starts the pattern-matching process. All pattern matching is limited to the temporal extent of the newly created ETNODE. The IEI algorithm first attempts to match DATERULEs found in the new ETNODE. A successful DATERULE recursively calls IEI to create a new child interval. After all dependent intervals that can be created have been instantiated, the IEI algorithm examines the FEATURERULEs. By executing DATERULEs before FEATURERULEs, the IEI algorithm ensures that all dependent contexts that can be instantiated are created before any FEATURERULE executes. With this property guaranteed by the algorithm, a FEATURERULE can contain predicates that test for the existence of dependent contexts within the temporal extent of the ETNODE, such as how many instances of a dependent context have been found. The rule-evaluation algorithm does not *require* that DATERULEs instantiate intervals or that FEATURERULEs not instantiate intervals; without this division of rule types, however, the IEI algorithm does not guarantee that all dependent intervals have been created when FEATURERULEs execute. Because DATERULEs and FEATURERULEs search for different types of patterns, the predicates found in these rule classes differ.

The IEI algorithm searches for patterns in data over time. In Figure 7.4, step 3 refers to an interval of time delimited by *date'* and *date''*. These dates specify the temporal interval over which the predicates in a RULENODE must hold for the predicate to return the value TRUE. ETNET rules use simple temporal-searching algorithms to assign values to *date'* and *date''* in the IEI algorithm. Two classes of temporal intervals, called *maximum spanning intervals* and *minimum spanning intervals*, are specified by the temporal-searching algorithm (Figure 7.5). A minimum spanning interval is the smallest interval over which a rule's predicates are always true. A rule with predicates that are true during some portions of a patient record will generate multiple minimum spanning intervals for each segment of the record that successfully matches the rule. Conversely, a maximum spanning interval is the largest interval in which the rule's predicates match at the interval endpoints but may not be true within the entire temporal extent of the interval. In minimum spanning intervals, the rule predicates must be true during the entire duration of the interval or else

IEI (*start*, *stop*, *type*, [*subtype*]):

1. Create ETNODE starting date = *start*; stopping date = *stop*
 2. Assert DATERULEs and FEATURERULEs for this type, [*subtype*]
 3. For *date* from *start* to *stop* do
 - (a) For rule in DATERULEs do
 - i. If Predicate(rule) = *true* on *date'* to *date''* then
 - A. $\text{type}' = \text{rule-type}$; $\text{subtype}' = \text{rule-subtype}$
 - B. IEI (*date'*, *date''*, *type'*, *subtype'*)
 - C. For actions in rule do Actions
 4. For rule in FEATURERULEs do
 - (a) If Predicate(rule) = *true* do Actions
-

Figure 7.4 The ETNODE instantiation algorithm.

The recursive IEI algorithm is invoked each time a new ETNODE is created. ETNODEs without rules have empty lists for DATERULEs and FEATURERULEs. The determination of values for *date'* and *date''* is presented in Section 7.2.3.

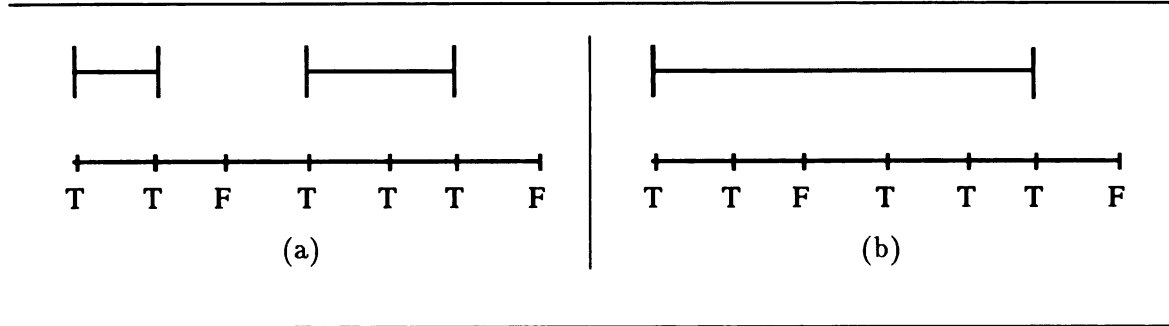


Figure 7.5 Minimum and maximum spanning intervals.

In this figure, T denotes a date during which the predicate is true; F denotes a date during which the predicate is false. A minimum spanning interval (a) is the smallest interval over which a rule predicate is always true. A maximum spanning interval (b) is the largest interval over which a rule predicate is true at the interval endpoints. In most cases, these intervals are not identical.

multiple minimum spanning intervals are created; in maximum spanning intervals, a rule with predicates that are true for some portion of a patient's record will create only a single maximum spanning interval, which represents the first and last time the rule's predicates are true.

With minimum and maximum spanning intervals, a total of four temporal searching strategies are available for evaluating RULENODE predicates. Figure 7.6 depicts the four strategies graphically; Table 7.1 presents the algorithms. These algorithms are responsible for assigning values to the terms *date'* and *date''* in the IEI algorithm (Figure 7.4). Two temporal-searching strategies (LL, RR) create minimum spanning intervals; two strategies (LR, RL) find maximum spanning intervals. Due to the construction of the IEI algorithm in TOPAZ, LL and RR rules create identical minimum spanning intervals. Likewise, LR and RL rules create identical maximum spanning intervals. In Section 7.4, I discuss a modification to the IEI algorithm that does not have this property.

RR rules start predicate matching at the left temporal bound of the ETNODE. Initially, this bound is the same as the ETNODE start date (Fig 7.4, step 3). If the predicate fails, the left temporal bound moves one date in chronological order and the predicate test is repeated. If the predicate succeeds, the RR algorithm continues to test the predicate on successive dates from the left bound in chronological order until the predicate fails. All dates within an interval created by an RR rule satisfy the rule predicate. LL rules are identical to

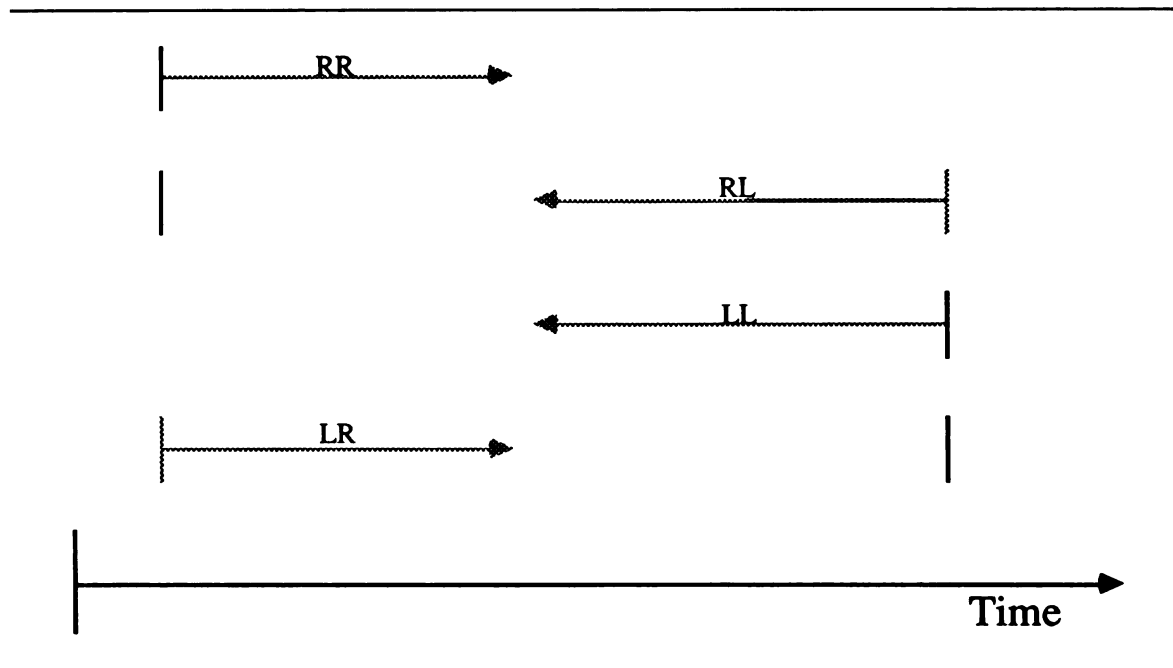


Figure 7.6 Temporal-searching diagrams.

This figure is a graphical depiction of the temporal-searching algorithms presented in Table 7.1. For each strategy, the solid vertical bar represents the first successful predicate match, and the dotted arrows represent the direction of additional predicate testing. For example, RR searches for the first instance of a successful predicate match. If such an instance is found, successive predicate testing occurs in chronological order until the predicate fails. Rules that specify RR temporal-searching create a minimum spanning interval, which represents the smallest interval of time during which the predicate was always true.

Table 7.1 Temporal-searching algorithms.

Symbol	Name	Action
RR	right moves rightward	If $P(\text{Left bound}) = T$ $date' = \text{left bound}$ $date'' = date'$ If $P(date'') = T$ $date'' = \text{next chronological date}$, else quit
RL	right moves leftward	If $P(\text{Left bound}) = T$ $date' = \text{left bound}$ $date'' = \text{right bound}$ If $P(date'') = T$, quit, else $date'' = \text{next reverse chronological date}$
LL	left moves leftward	If $P(\text{Right bound}) = T$ $date'' = \text{right bound}$ $date' = date''$ If $P(date') = T$, $date' = \text{next reverse chronological date}$, else quit
LR	left moves rightward	If $P(\text{Right bound}) = T$ $date'' = \text{right bound}$ $date' = \text{left bound}$ If $P(date') = T$, quit, else $date' = \text{next chronological date}$

RR rules except that they begin predicate matching at the right temporal bound (initially equal to the ETNODE stop date) and, if successful, predicate testing proceeds from the right bound in reverse chronological order until the predicate fails. Like intervals created by RR rules, intervals created by LL rules have the property that all dates within the interval satisfy the LL rule predicates. RL rules first test the predicate at the left bound. If the predicate is true, then, starting from the right bound in reverse chronological order, an RL rule does predicate matching on each date and terminates on the first date that satisfies the predicate. If the predicate is false at the left bound, the left bound moves one date in chronological order and the predicate testing is repeated. LR rules first test the predicate at the right bound. If the predicate is true, then, starting from the left bound in chronological order, an LR rule does predicate matching on each date and terminates on the first date that satisfies the predicate. If the predicate is false at the right bound, the right bound moves one date in reverse chronological order and the predicate testing is repeated. An interval created by either an RL rule or an LR rule satisfies the predicate at its endpoints but may not satisfy the predicate at all dates within its temporal extent. RR rules and LL rules create minimum spanning intervals, where all dates within the interval satisfy the rule predicate. Conversely, RL rules and LR rules create maximum spanning intervals, where the predicate is satisfied at the most extreme two endpoint dates, but may not be true on dates within the interval's temporal extent.

A RULENODE can declare only one of these temporal-searching strategies. The value of the TIMEDIRECTION slot in a RULENODE is LL, LR, RL, or RR. This value specifies which strategy the temporal pattern-matching procedure is to use when evaluating this RULENODE. The four temporal-searching strategies, combined with the IEI algorithm, provide the knowledge engineer with a wide range of options for temporal pattern matching. The choice of minimum or maximum spanning rules is a knowledge-acquisition decision. TOPAZ uses each of the four temporal-searching strategies in at least one rule. Although other searching strategies could be devised, the four presented in Figure 7.6 and in Table 7.1 are sufficient to represent complex patterns in TOPAZ.

7.3 The TOPAZ Temporal Model

TOPAZ represents remarkable clinical events using ETNODEs (Figure 7.7). It infers these events using RULENODEs. TOPAZ searches for two types of temporal ab-

Table 7.2 Pair-wise comparisons for summarization and explanation.

Summarization concept	Comparison used to derive concept	Notation
surprising observations	population-based model predictions vs patient-specific model predictions	Pred.v.Pop
explanatory states	population-based model parameters vs patient-specific parameter estimates	PopP.v.PredP
model critique	patient observations vs patient-specific model predictions	Obs.v.Pred

stractions: (1) intervals of *surprising* or *unusual* clinical observations, and (2) intervals of *potential* misinterpretation that may result in erroneous or conservative conclusions. In this section, I describe how TOPAZ exploits the patient-specific bone-marrow model (Chapter 6) and ETNET to derive temporal abstractions of remarkable clinical events.

At any time during the patient’s clinical course, the bone-marrow model generates predicted WBC counts ($\hat{y}_i$). When supplied with the population parameter values (ϕ_0), the model predicts the expected WBC counts for a typical or average patient receiving MOPP chemotherapy. When supplied with patient-specific parameters ($\hat{\phi}$), the model predicts the expected WBC counts for a specific individual receiving the same chemotherapy. TOPAZ examines pair-wise differences among patient observations (y_i), population-based predictions ($y_i(\phi_0)$), and patient-based predictions ($y_i(\hat{\phi})$) to infer meaningful temporal abstractions. For example, if we assume our patient is typical, then a large difference between his patient-specific predicted WBC count and the population-based predicted WBC count is surprising (Table 7.2). TOPAZ creates an ETNODE to denote any interval of time during which large deviations from the expected clinical course were observed (Figure 7.8). The domain expert specifies how large a difference must be seen for the difference to be considered surprising or atypical. Once this threshold is exceeded, TOPAZ considers each episode to be notable, significant, and worthy of comment and explanation. Each ETNODE generated at this step contains rules that then search for possible explanatory causes of this significant deviation.

The patient-specific model provides more than predicted WBC counts. It also encodes the state of knowledge about the underlying physiological processes in this individual in the estimated model parameters ($\hat{\phi}$, Table 6.2) and in the predicted model states ($\mathbf{z}$, Ta-

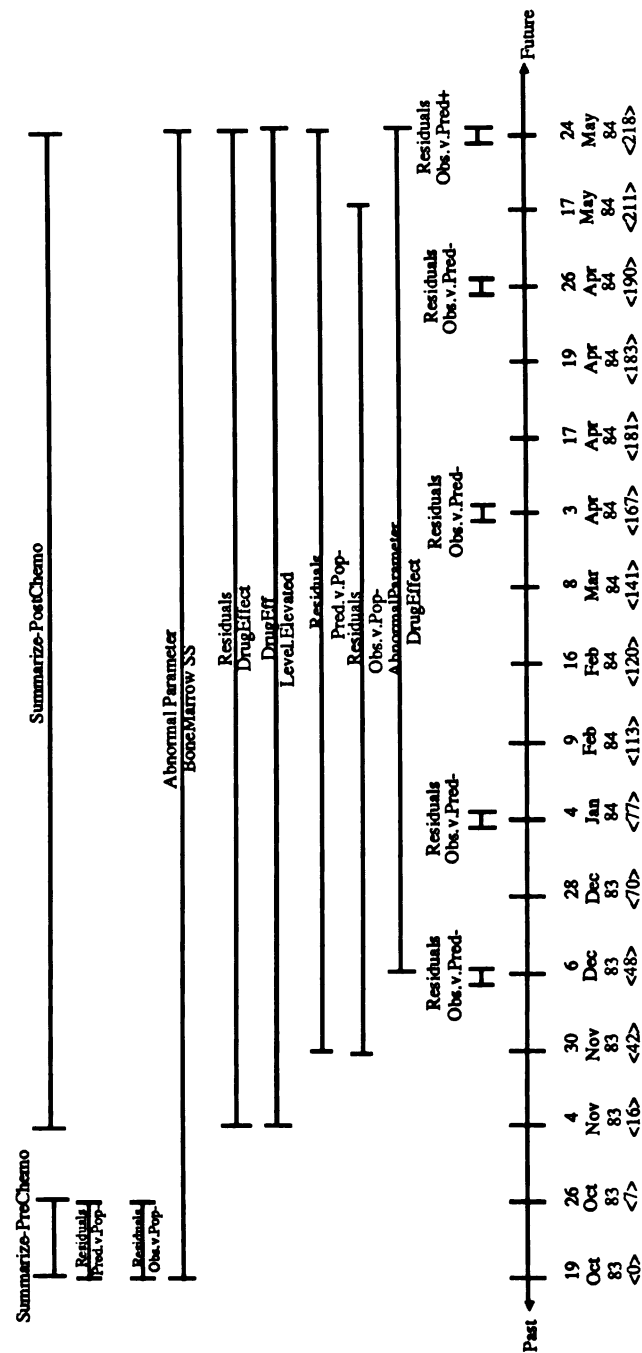


Figure 7.7 Clinical abstractions represented in ETNET.

Notable differences among the patient observations, population-based expectations, and patient-specific model predictions trigger the creation of ETNODEs. These ETNODEs can create other abstraction intervals that attempt to critique the abstraction and to explain the differences. The set of ETNODEs created by TOPAZ is used by the text-generation module (Chapter 8) to produce the final summary text.

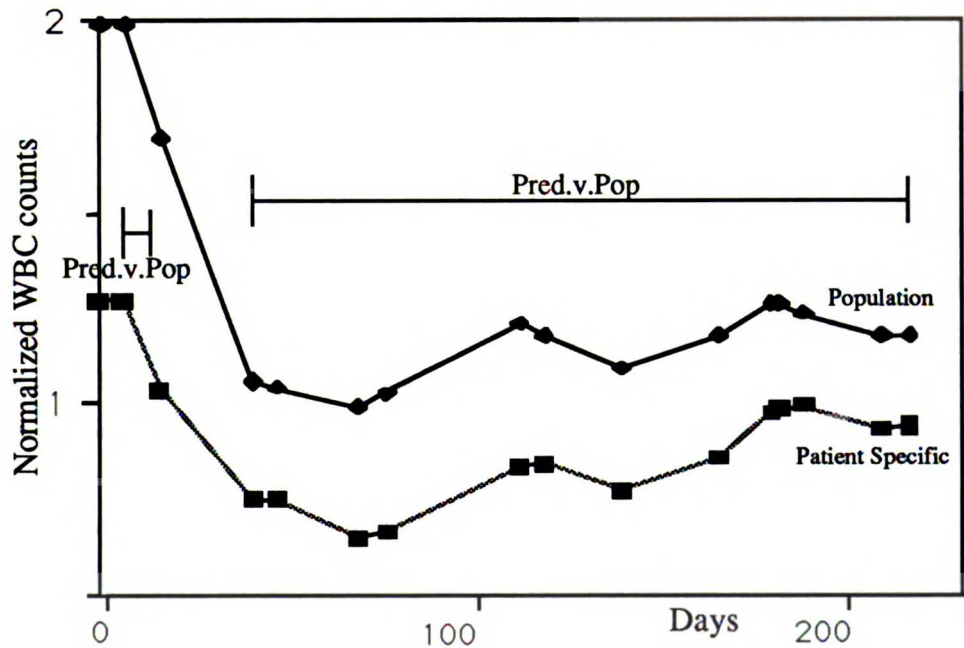


Figure 7.8 ETNODE intervals created by model prediction differences.

Large differences between patient-specific predictions and population-based predictions causes the creation of an ETNODE with a Pred.v.Pop type-label. An ETNODE of this type represents an interval of time during which the patient's modeled clinical course differs from the expected clinical course. Rules associated with Pred.v.Pop ETNODEs attempt to find an explanation for this discrepancy. Other rules attempt to critique the conclusion. In this figure, the dotted line plots the patient-specific model predictions, the solid line plots the population-based model predictions, and the intervals denote differences of sufficient magnitude to cause the creation of a Pred.v.Pop ETNODE. The data for this figure is from the patient record shown in Figure 5.6.

ble 6.3). For model fitting (Section 6.3), only the model predictions for granulocyte counts are used. However, the model state vector, with six elements, has predicted values for other modeled physiological features, such as the bone-marrow and the drug-effect compartments. The population-based model calculates values for these same physiological states, but these predicted states are the expected values for a typical patient. A change in a patient-specific estimated parameter value reflects an alteration in the modeled physiological process that is required for the model output to predict the actual observations. Differences in the underlying states predicted by these two models provide a method to explain differences in the predicted WBC count, which gives a physiological-based explanation for the unusual observations.

TOPAZ creates an ETNODE for each interval where there are large differences in predicted granulocyte counts. For each unusual event, ETNET rules examine the underlying model states and estimated parameters for potential physiological causes for the discrepancy (Figure 7.9). These intervals represent underlying model states that can cause unusual granulocyte counts. TOPAZ formulates a comment in the final patient summary for each ETNODE created in this analysis.

The patient-specific model simulates the patient's true clinical course. Because the model is incomplete and the data are error-prone, there are differences between the model's predicted granulocyte counts and the actual measured granulocyte counts. Small differences between the patient-specific model and the patient observations are not worrisome, but large systematic differences signify a problem with the model fit. Using the same approach for differences between population-based model predictions and patient-specific model predictions, TOPAZ compares differences between the patient-specific model predictions and the patient observations to examine the ability of the model to explain the patient's clinical course. TOPAZ creates an ETNODE, with type label `Obs.v.Pred`, for each interval where there are large differences between the patient-specific model prediction and the actual patient observation (Figure 7.10). A single observation that is poorly fit by the model implies a less serious problem with the model conclusions than does a series of consecutive poorly fitted observations. TOPAZ considers any episode of two or more consecutive observations that are poorly predicted by the patient-specific model to indicate systematic model error. For each interval of this type, FEATURERULEs examine the size and direction of the discrepancy to determine what effect the difference has on the final analysis. This examination provides a method to critique the abstractions generated by

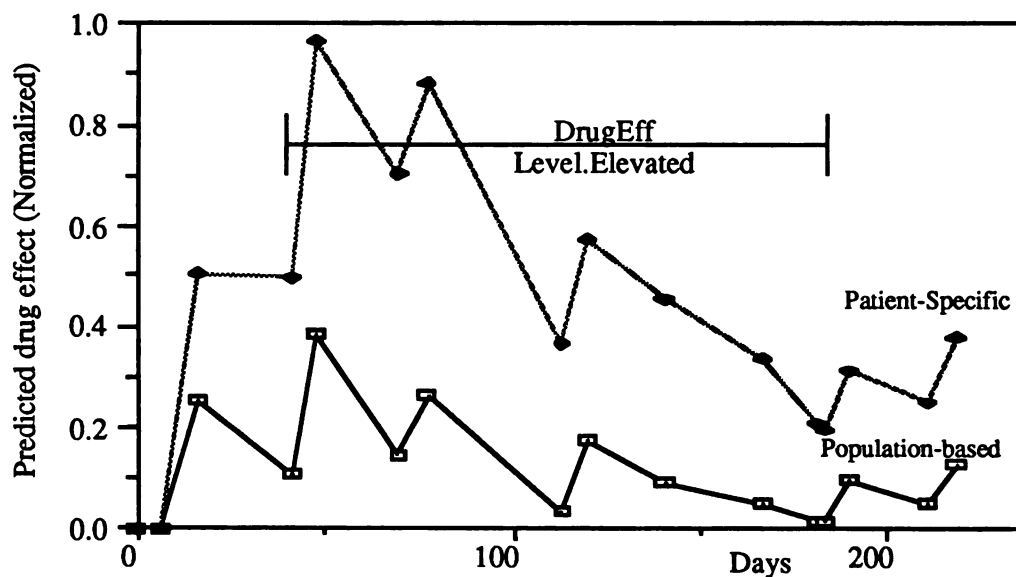


Figure 7.9 Causal explanations of surprising observations.

Differences between patient-specific model WBC predictions and population-based model WBC predictions create an ETNODE containing rules that look at predicted model states for underlying physiological states that can explain the discrepancy (Fig 7.8). Both the patient-specific and population-based bone-marrow models also simulate changes in other compartments, such as the bone-marrow and drug-effect compartments. Differences in these compartments can cause the observed WBC differences, which provide a causal explanation to be generated in the final summary. The dotted line plots the patient-specific model predictions of the drug-effect compartment and the solid line plots the population-based model predictions of the same compartment. These plots are from the same patient whose predicted WBC counts are graphed in Figure 7.8. Differences in drug-effect explains the second Pred.v.Pop ETNODE but not the first ETNODE.

predicted model differences. If no Obs.v.Pred intervals are created, the model predicts the observations well over this interval. If an Obs.v.Pred interval is created, then the model either under- or over-predicts differences between WBC counts over this interval.

A property of Bayesian estimation is that the parameter estimates tend to be biased toward the prior estimates. This bias leads to final estimated parameter values that are more conservative than they would be with a non-Bayesian estimation technique. TOPAZ notes whether the difference between the patient-specific model predictions and patient observations signals an overly conservative conclusion that would be strengthened by a better fit, or whether the difference signals a potentially erroneous conclusion that might be invalidated if a better model fit were available. This analysis appears in the final TOPAZ summary.

7.4 Problems and Extensions

The model of temporal abstraction presented in the previous section is composed of two related parts: (1) the domain-independent ETNET system that constructs interval abstractions and (2) the domain-dependent rules that define interesting model-based abstractions for summarization. Although these two components of temporal abstraction are intimately linked, I examine these two components of the TOPAZ temporal model separately in this section.

7.4.1 The ETNET system

ETNET is a knowledge-representation technique, a temporal-reasoning mechanism, and a database-management system. As a knowledge-representation scheme, ETNET encodes temporal abstractions as intervals of time during which some event is occurring. As a temporal-reasoning system, ETNET adds and removes rule sets as intervals are created and terminated. As a database-management system, ETNET intervals are access paths for context-dependent data storage and retrieval. In this section, the limitations of ETNET and areas for future work are examined from these three perspectives.

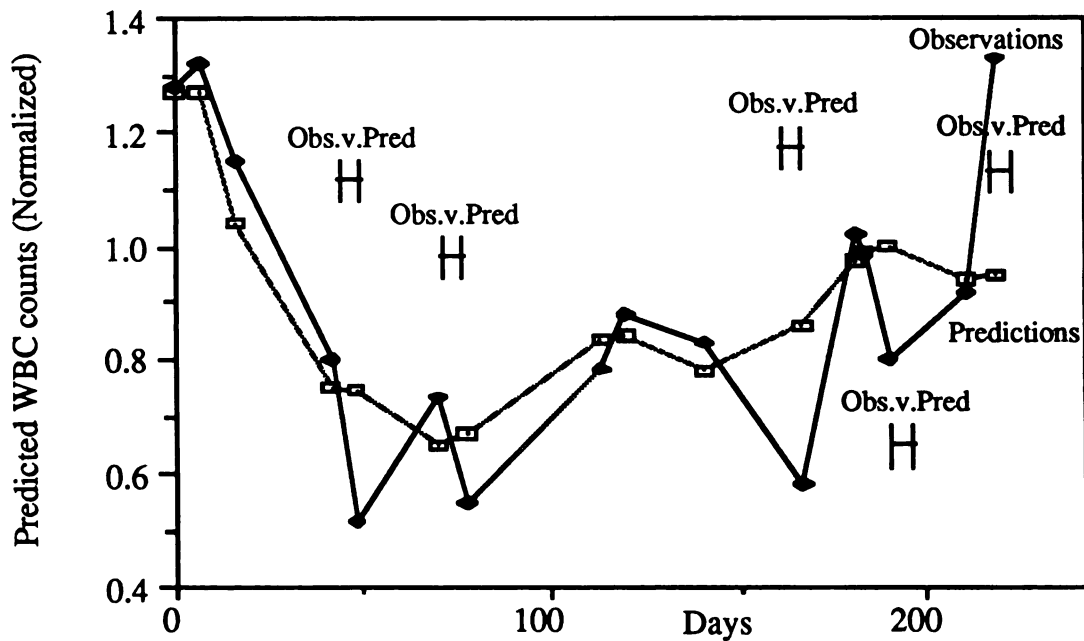


Figure 7.10 Critiquing the patient-specific model fit.

Large differences between observed WBC values and patient-specific model predictions create an ETNODE with type label Obs.v.Pred. This node represents an interval of time during which the modeled patient differed from the observed clinical course. During these intervals, abstractions derived from the patient-specific model may be excessively conservative or potentially erroneous. TOPAZ generates a critique of the model conclusions by examining the size and direction of the discrepancies during these intervals. The dotted line plots the predicted WBC counts generated by the patient-specific model, the solid line plots the observed WBC counts for the same patient as shown in Figure 7.8.

7.4.1.1 The Representation of Temporal Knowledge using ETNET

ETNET is a generalization of the ONCOCIN TNET (Chapter 4). The two systems share a common strong assumption: Interval endpoints must be unambiguous. For all ETNET objects, the starting (creation) and stopping (termination) dates must be a known calendar date. Only two “dates” are exceptions to this rule: (1) *future* is defined as sometime after the latest calendar date in the current network, and (2) *past* is defined as sometime before the earliest calendar date in the current network. ETNET cannot encode dates that are imprecise (e.g., “last month”), uncertain (e.g., “possibly last year”), or disjunctive (e.g., “either today or tomorrow”). ETNET does support changing an interval’s starting or stopping time to a new date if additional information causes the current interval times to be incorrect; in all cases, however, the dates must be specified completely.

The assumption of unambiguous dates is fundamental to the design and implementation of ETNET. It is not possible to suggest a simple modification that would remove this restriction. Because of this assumption, ETNET is not appropriate as a knowledge-representation scheme for certain important temporal-reasoning tasks. ETNET cannot encode the uncertain temporal relationships that appear in a patient’s past medical history.¹ Also, ETNET is not appropriate for therapy-planning systems that attempt to model potential future clinical events whose starting and ending times cannot be precisely predicted by the planning system.

7.4.1.2 Temporal Reasoning

The ability of ETNET to detect temporal patterns is determined by the expressiveness of the rule predicates and by the flexibility of the searching algorithm. In TOPAZ, I constructed a rudimentary set of rule predicates (Appendix B). A simple set of predicates is adequate in TOPAZ because the patient-specific bone-marrow model provides sufficient constraints on the problem that only simple temporal pattern making is required. In a situation where more complex patterns might need to be represented within ETNET, a larger set of rule predicates may be required.

The IEI temporal-searching algorithm (Figure 7.4) limits the class of temporal patterns that TOPAZ can detect. The strongest constraint is imposed by the restriction of temporal pattern matching to within the temporal extent of the parent ETNET interval

¹See the discussion of Kohane’s THRIPTH in Section 2.2.1.2.2.

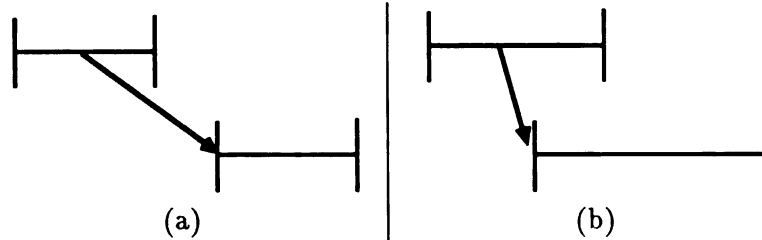


Figure 7.11 Temporal dependencies not detected by the temporal-searching algorithm.

The IEI temporal-searching algorithm (Figure 7.4) prevents ETNODE rules from detecting dependent intervals such as the two examples shown in this figure. The illustration in (a) depicts a temporally dependent event that begins after the enabling or conditioning event terminates. The IEI algorithm limits pattern matching to the temporal extent of the conditioning event only. The illustration in (b) depicts a temporally dependent event that persists after the enabling event terminates. The IEI algorithm would terminate the dependent event at the same time that the enabling event terminates.

(*date'* and *date''* must be within the interval defined by START and STOP). Two classes of contextually dependent events cannot be detected because of this restriction: (1) events that begin after the parent event terminates, but are dependent on the existence of the parent event as an enabling or predisposing prior event; and (2) events that begin during the parent event, but persist after the parent event terminates (Figure 7.11). The inability to detect these intervals is a limitation of the temporal-searching algorithm, not a limitation of the representational ability of ETNET to model the existence of this type of interval. There are nonintuitive techniques to coerce the inferencing system to detect these events, but the IEI algorithm must be changed to express these patterns in a straightforward manner. One side effect of this change would be that LL and RR rules would not create identical minimum spanning temporal intervals and LR and RL rules would not create identical maximum spanning temporal intervals. The main disadvantage of generalizing the temporal-searching routine is that the computational complexity quickly increases as more potential combinations of ETNODEs must be examined by rule predicates.

ETNET infers oscillations, cycles, and other periodic events in an awkward manner. Each oscillation or cycle triggers the creation of a new ETNODE denoting the start and stop of that oscillation (or cycle). Higher abstraction intervals aggregate a period of successive cycles into a single ETNODE denoting the oscillatory (or cyclic) period (Figure 7.12).

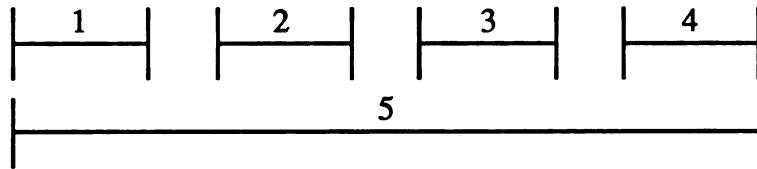


Figure 7.12 Abstraction of cyclic events in ETNET.

ETNET abstracts cycles by aggregating multiple episodes of similar events into a single encompassing abstraction interval. For this abstraction to occur, each episode must be detected individually before the aggregating abstraction is created. This approach is both awkward and inefficient. In this figure, the number on each interval indicates the order in which that interval is created by ETNET.

This technique is inefficient and cumbersome. It arises from a lack of a pattern-matching scheme that can express recurring patterns directly. This is a restriction in both the rule-predicate language and the temporal-searching algorithm.

7.4.1.3 Database Management

In generalizing TNET to ETNET, our ability to find temporally dependent contexts without search was lost. A *temporally dependent context* is an interval that must start and stop wholly within the temporal extent of another interval. TNET maintains complex lists of temporally dependent intervals within the TCHILDREN space of each TNODE (Chapter 4). These lists were eliminated in ETNET to increase the representation's flexibility to add and delete intervals (ETNODEs) anywhere within a temporal network. In ETNET, to find all instances of context A that are temporally dependent on context B, ETNET must serially examine all dates between the starting and stopping dates of context B for the presence of an instance of context A that starts and stops within context B. Large intervals, extending over numerous dates, can potentially cause substantial performance degradation because of the time it takes to perform the linear search. In actual usage, this potential problem has not arisen, but I would expect degraded performance to become significant in complex temporal settings.

Both TNET and ETNET do not contain the database-management facilities found in most commercial products. Important DBMS features not implemented in either system include data-integrity checks, data-consistency constraints, concurrency and updating

control, and secondary-storage access methods. Rather than extend the capabilities of ETNET to encompass these traditional DBMS features, I believe it would be more efficacious to reimplement ETNET using a full-fledged DBMS that contains these missing features. Figure 7.13 illustrates a two-table relational schema that can represent an ETNODE. The relation TNODE encodes the fields found in the TNODE space in an ETNODE (Section 7.2); the relation TRESULT encodes the fields found in the TRESULT space. The TNODE relation is a direct translation of the fields found in the object-based implementation; the TRESULT relation requires more explanation. In ETNET, a parameter (PNAME) may consist of a single item or of a list of items. Because the relational-data model requires the fields of all tuples to contain only atomic values, the TRESULT relation contains an ITEM# field to separate multiple values for the same parameter. Also, ETNET does not replace recomputed or reasserted database values. Rather than destroying old values, ETNET maintains all previously asserted values for a parameter, but it returns only the most recently asserted value in response to a query request. To reproduce this feature in a relational design, the TRESULT relation has an ASSERT# field that is incremented each time a new value (or values for a multivalued attribute) for PNAME is asserted. Note that this recasting of ETNET into a two-table relational DBMS schema does not require the representation of the TOPNODE object described in Section 7.2. This simple implementation makes three key assumptions: (1) a dating scheme exists in the commercial relational DBMS's data manipulation language such that, if *date1* precedes *date2*, then $date1 < date2$; (2) two unique dates, *past* and *future*, can be declared such that $\forall d \in dates ((PAST < d) \wedge (d > FUTURE))$; and (3) for each parameter, values can have one and only one predefined type (for example, only integer, real, or string values).

The recasting of ETNET into a traditional relational DBMS destroys the ability to associate inferencing rules with temporal intervals. POSTGRES, a proposed extension of INGRES [117], has been designed to include rules as valid database objects. This extension to the traditional relational model may provide a mechanism to include the temporal-reasoning features as well as the temporal-representation features of ETNET within a commercial relational DBMS.

RELATION: TNODE	RELATION: TRESULT
ID# (integer) ::	ID# (integer)
START (string)	PNAME (string)
STOP (string)	Assert# (integer)
TYPE (string)	Item# (integer) ::
SUBTYPE (string)	Value (user-defined)

Figure 7.13 A two-table relational scheme to encode an ETNODE.

The ETNODEs objects in ETNET can be reimplemented as a two-table relational database. The relation TNODE represents the fields in the TNODE object space; the relation TRESULT represents the fields in the TRESULT object space. In this figure, the key attributes in each relation precede the double colon, the nonkey attributes follow the double colon. For each attribute, the corresponding data type is listed in parentheses.

7.4.2 Model-Based Temporal Abstraction

The TOPAZ context model uses differences among patient observations, patient-specific model predictions, and population-based model predictions to define relevant concepts such as an *unusual clinical event* or a *poorly understood (modeled) patient observation*. The context model also relies on the components of the structural model, such as estimated parameter values, to formulate a causal explanation for each unusual event. A central thesis of this dissertation is that an explicit model that encodes an understanding of the relationships between inputs and outputs (for example, the TOPAZ bone-marrow model is an explicit model, with the inputs being chemotherapy drugs and the output being WBC counts) provides intuitive definitions for clinically useful semantic concepts. In this section, I examine the limitations that can arise from this approach to summarizing time-ordered clinical data.

TOPAZ defines an unusual event whenever there is a large difference between the patient-specific model predictions and the population-based expected clinical course. Notice the phrase, *After adjusting for the patient's steady-state bone-marrow mass*, in the summary shown in Figure 5.7. This statement indicates that the expected clinical course was modified after the initial clinical data (prior to the patient receiving any chemotherapy) indicated a lower-than-expected pretreatment bone-marrow mass. In this case, the expected clinical course is no longer strictly the population-based predicted clinical course,

but rather is the predicted course after adjusting for this patient’s prechemotherapy steady-state bone-marrow mass. Failure to make this adjustment results in a single interval, covering the entire patient record, that is considered highly unusual (Figure 7.14). TOPAZ uses the population-based *prototypical* model to generate expected predictions during the prechemotherapy phase, and a *static patient-specific* model (PSM) (Figure 5.1) to generate expected predictions during the chemotherapy portion of a patient’s clinical course. In Figure 7.7, the ETNODE SUMMARIZE-PRECHEMO represents the interval during which the prototypical model was used as the expected value, whereas the ETNODE SUMMARIZE-POSTCHEMO represents the interval during which the static PSM was used as the expected clinical course. The patient-specific predictions change as new patient information become available, but here TOPAZ also changes the “population-based” predictions based on patient data. Changing the model used to generate the expectations is stretching the meaning of the term *population-based expectations*. It makes sense that, after the first WBC count is observed, we know more about which subpopulation of patients with Hodgkins’ lymphoma best describes this patient (e.g., the subpopulation of patients with hypercellular bone marrows versus the subpopulation of patients with normal bone marrows). This information cannot be derived from other patient characteristics. Dividing the clinical course into prechemotherapy and postchemotherapy phases also makes sense because the two contexts are substantially different. It is useless to look for elevated drug effect to explain a lower-than-expected pretreatment WBC count. Likewise, it is incorrect to suggest that a lower-than-expected bone-marrow mass is the explanation for a lower-than-expected WBC count during chemotherapy if the expected predicted value already factors out this feature. TOPAZ readily handles this situation by using different rule sets that are contextually dependent on SUMMARIZE-PRECHEMO and SUMMARIZE-POSTCHEMO ETNODEs.

The presence of critical contexts that may require a change to a new “population-based” expected clinical course raises a significant question: Can domain-independent criteria that can be applied to a wide range of dynamic models be defined that would identify when a new set of expectations should be invoked? I used clinical knowledge to recognize that some patient records were being summarized as one long abnormal event, due to the failure to adjust the patient’s expected clinical courses in light of a low initial bone-marrow mass. I also used clinical knowledge to justify the separation of prechemotherapy and postchemotherapy contexts and to use different models of the expected clinical course during these intervals. As demonstrated by TOPAZ, always using the true population-based

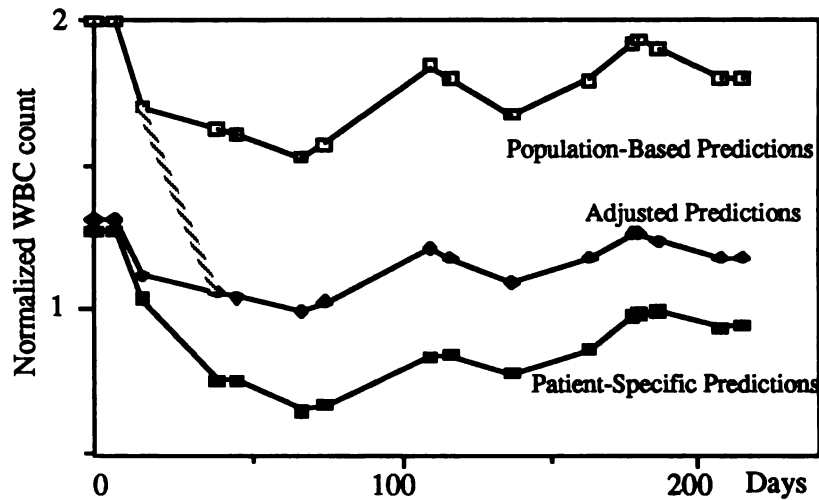


Figure 7.14 Patient-specific alterations to the expected clinical course.

A lower-than-expected pretreatment WBC count indicates a lower-than-expected pretreatment bone-marrow mass. Failure to adjust the expected clinical course to account for this patient-specific feature results in a single abnormal event covering the entire patient record. This figure plots the unadjusted population-based predicted WBC counts, the predicted WBC counts after adjustment of the population-based model for the low pretreatment bone-marrow mass, and the patient-specific predicted WBC counts. If the adjustment for the low bone-marrow mass were not made, a single abnormal interval would be created; with the adjustment made, two intervals (one before the start of chemotherapy and one after the start of chemotherapy) are created (see Figure 7.8). The plots shown in this figure are unlike those in Figure 7.8. In Figure 7.8, the first part of the “expected” clinical course are population-based predictions, whereas the second part of the “expected” clinical course are pretreatment adjusted predictions. In this figure, the same change to a different “expected” clinical course is marked by the thick dotted line from the population-based plot to the adjusted plot.

expectations does not make sense if an early observation changes the expectations for all subsequent predictions. A strictly enforced definition of the *expected clinical course* is not feasible, yet TOPAZ makes strong assumptions about the semantics of the “expected clinical course” when it compares the expected predictions with the patient-specific predictions and with the patient observations.

A similar semantic problem arises in defining the patient’s *true* clinical course. TOPAZ has at least two candidates to use as the true clinical course—the actual patient observations and the patient-specific model predictions. In detecting unusual clinical events, I use the latter. Therefore, in Table 7.2, TOPAZ detects surprising observations by comparing the population-based model predictions not with actual patient data, but rather with patient-specific model predictions. This choice may appear counterintuitive, because I do not use the actual clinical measurements to describe the patient’s clinical course. All clinical observations contain significant random measurement error, which means that wide swings in measurements are not unusual. Using patient observations rather than patient-specific model predictions to detect unusual clinical events results in the generation of many erroneous TOPAZ intervals (Figure 7.15). Due to the physiological constraints encoded in the bone-marrow model and due to the averaging effect caused by the inclusion of all patient observations in the model fit, the patient-specific model predictions are smoother than are the observed values. Because the model encodes extensive knowledge about the fundamental relationships that exist in the domain, I use the model predictions rather than the actual observations to represent the best current understanding of the patient.

The actual patient observations and the patient-specific model predictions are not the only choices that are available to us for defining the patient’s clinical course. The patient-specific model can be viewed as a highly constrained, parametric smoothing function, which modifies (smooths) the original observations based on the physiological relationships encoded in the model structure. From this perspective, the patient observations can be viewed as unsmoothed data. By using the physiological model to smooth the observations, I eliminate large swings in the data, but true changes in the data are also damped. A continuum of nonparametric and parametric smoothing techniques exist that could be used to generate smoothed data that are less vulnerable to the large errors of unsmoothed clinical data but are more permissive to true measurement changes than is the highly constrained physiological model. A data-driven, nonparametric smoothing procedure that uses only data that are nearby the point to be smoothed would be substantially less computationally

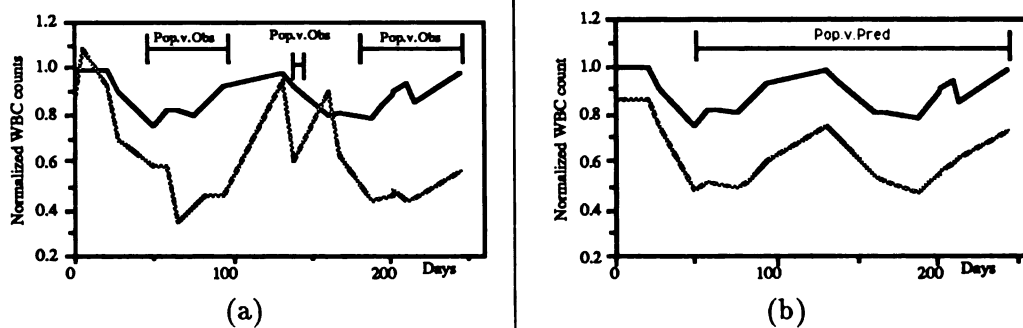


Figure 7.15 Patient observations versus model predictions to define the true clinical course.

In (a), the expected clinical course (solid line) and the patient observations (dashed line) are plotted. Three intervals of large differences (labeled Pop.v.Obs) are created. In (b), the expected clinical course (solid line) and the patient-specific model predictions (dashed line) are plotted. Only one interval of large differences (labeled Pop.v.Pred) is created. The three smaller intervals in (a) are caused by two spurious observations that were sufficiently close to the expected value to terminate the Pop.v.Obs interval. The patient-specific model predictions (b) are smoother than are the patient observations (a) due to the constraints encoded in the model and averaging effects from the fitting procedure. The smoothing behavior of model fitting eliminates these two sharp spikes in the observed data, resulting in only one abnormal interval. The data for these plots were not from the example patient summarized in Figure 5.7. The plot in (b) in this figure is analogous to Figure 7.8 for the example patient.

Table 7.3 Pair-wise comparisons for summarization using data smoothing.

Summarization concept	Comparison used to derive concept	Notation
surprising observations	population-based model predictions vs nonparametric smoothed data	Pop.v.Smooth
explanatory states	population-based model parameters vs patient-specific parameter estimates	PopP.v.PredP
model critique	nonparametric smoothed data vs patient-specific model predictions	Smooth.v.Pred

expensive than is calculating the best bone-marrow parameters for the complete set of patient data. If this approach were adopted, Table 7.2 would be modified to Table 7.3. This alteration in detecting surprising observations could improve the performance of TOPAZ in a real-time data-acquisition environment. For example, in the current model-based fitting approach, new data require a new parameter fit before a model prediction can be generated and thus can be compared to the expected value. A data-directed local smoothing procedure that was computationally inexpensive could be used to monitor incoming patient data for surprising observations. The model-fitting routines would be invoked only when the patient demonstrated a deviant clinical course. TOPAZ would need to perform a complete (expensive) model fit to explain the unusual observation only when the smoothed data were sufficiently surprising to warrant it.

A potential problem caused by using the patient-specific model predictions as the patient's true clinical course rather than the actual patient observations is that these two vectors are not identical ($Y \neq \hat{Y}$). Only the values in Y appear in the medical record; the values in $\hat{Y}$ are derived numbers. TOPAZ creates intervals of surprising clinical events based on numbers that are not present in the patient's medical record. In settings where the actual measurement is markedly different from the model prediction, the user does not know why TOPAZ seems to ignore a highly abnormal measurement. This point is resolved later in the summary paragraph. When TOPAZ critiques the fit used to generate the summary, it will note that there was an observation that was poorly fitted by the patient-specific model. This potential confusion is a direct side effect of using data other than the actual observations, such as the patient-specific model predictions or a smoothing procedure, as the patient's true clinical course. The only alternative is to use the patient observations

directly and to encode some model of the expected errors that would permit TOPAZ to ignore isolated or nonsystematic unusual observations.

7.5 Summary

In this chapter, I have presented the interval-based abstraction model used in TOPAZ. This model is implemented ETNET, a symbolic system for temporal-reasoning and temporal-representation. TOPAZ abstractions are intervals generated by context-sensitive rules in ETNET nodes. These rules examine features of the bone-marrow model predictions, bone-marrow model parameters, and patient observations to derive abstractions over time. The effect of ETNET is to convert the process-oriented view of time embedded in the bone-marrow model to an event-oriented view of time embedded in the intervals represented by ETNODEs. A physician who is well trained in mathematical reasoning would probably find this transformation unnecessary. Given the details of the underlying model, the parameter values, the model predictions, and medical knowledge about the modeled domain, he has enough information to understand fully the TOPAZ analysis of the patient's clinical course. ETNET abstractions do not add any additional analytic information to the final summary. ETNODEs aggregate and organize numbers generated by the process model into related contexts. In TOPAZ, the interval-based abstraction model encodes the context-specific, medical knowledge that attaches clinical relevance and importance to the numbers produced by the bone-marrow model. In addition, the TOPAZ abstraction model contains knowledge about the bone-marrow model structure. This knowledge is used to find plausible causal explanations for detected unusual events. Finally, the TOPAZ abstraction model contains knowledge about how to critique conclusions based on the bone-marrow model fit. All three forms of knowledge are found in the summary analysis generated by TOPAZ. In the next chapter, I shall describe how a state-based model of text generation converts ETNET abstractions to medical prose.

CHAPTER 8

A State-Based Model of Summary-Text Generation

TOPAZ summary text is designed to mimic the style of a physician describing to a colleague a patient whose course is complicated. These conversations, which frequently occur in the hospital hallways, are a quick synopsis of the key points about a patient. Details are expressed if they are germane to the main issue; otherwise, they are not mentioned. When focused on the patient's clinical measurements, physicians discuss more than just the numbers; they also interpret what these numbers mean clinically.

In Chapter 5, I presented the general structure of a clinical summary. A summary has two key parts: (1) a description of the key unexpected clinical events, and (2) an analysis of the clinical implications of these unusual events. Although this overall structure is useful to segregate certain topics into main paragraphs, the content of each paragraph is both patient- and problem-specific. The patient record determines which features are present; the inferred problems determine which features are relevant. The text-generation system should alter its contents according to both criteria. In this chapter, I shall describe the underlying model of summary-text generation embodied in TOPAZ. I shall describe the technical details of the TOPAZ text-generation system. As in previous chapters, I shall use the patient summary shown in Figure 5.7 in all examples.

Table 8.1 The TOPAZ organizational schema.

Discourse topic	Summary content
patient-identifying data	set the overall clinical context
analysis of clinical course	present the key implications of the clinical course; interpret the key clinical events in light of these implications

TOPAZ generates summary text using an explicit model of clinical presentation. This model encodes the style and content of a quick clinical synopsis of a patient's clinical course. For the summary to be a true synopsis, TOPAZ must be able to present patient details only if they are germane to the main patient problems; otherwise, these details are suppressed. In Chapter 5, I decomposed the TOPAZ presentation model into two presentation schemas: (1) the organizational schema that determines the overall logical structure of a summary, and (2) a rhetorical schema that encodes the logical components of a clinical argument. Table 8.1 presents the organizational schema encoded in TOPAZ; Figure 8.1 outlines the rhetorical schema. The main purpose of the organizational schema is to provide overall coherence and structure to the summary text. Physicians are accustomed to finding clinical information in a certain order. For example, the patient-identifying data always precede any discussion about patient findings, so that the general clinical context is set for the remainder of the presentation. The rhetorical schema is more complex. It must encode a model of cogent and convincing clinical debate. The rhetorical schema is responsible for selecting features from the analysis that support a summary statement. Because there may be conflicting evidence present in the medical record, the rhetorical schema must be able to both support and critique each conclusion reached by TOPAZ.

The purpose of the text-generation system is to convert abstraction intervals represented as ETNET intervals into an output format acceptable to clinicians. In the next section, I describe the state-based text-generating networks that encode both the organizational and the rhetorical schemas in TOPAZ.

-
1. Does the patient-specific model explain the remarkable clinical event?
 - Yes: Use the model to provide a causal explanation.
 - No: Is the unexplained remarkable clinical event nonsystematic (duration only one observation)?
 - Yes: Assume it is a transient factor not encoded in the model.
 - No: Warn about a potential systematic deficiency in the current patient-specific model.
 2. How well does the model predict the remarkable clinical event?
 - Acceptable: No comment.
 - Unacceptable: Is the model potentially conservative?
 - Yes: Remark that model estimates probably are conservative despite poor predictions.
 - No: Remark that model estimates probably are extreme.
 3. How well does the model predict the remaining clinical course?
 - Use the same logic as step 2.
-

Figure 8.1 The TOPAZ rhetorical schema.

Each conclusion reached by TOPAZ is examined by the logic encoded in the rhetorical schema. The schema ensures that all conclusions presented to the reader also contain a critique of the degree to which the patient-specific model was able to predict the finding.

8.1 Augmented Transition Networks for Text Generation

The TOPAZ presentation model is implemented as a hierarchy of augmented transition networks (ATNs; Figure 8.2). An ATN is a collection of nodes connected by directed arcs. Nodes contain logical statements that determine which arc is to be traversed to reach the next node to be processed. A node or arc can contain actions that are executed only if that node or arc is traversed. ATNs have two features that are useful for prose generation: (1) nodes can be organized into hierarchical units, and (2) intermediate results can be shared among nodes and passed between network hierarchies. An ATN hierarchy results when one of the actions performed by an ATN node is the execution of another ATN. When the SUBATN concludes processing, the calling ATN continues with its processing. ATN hierarchies are useful for decomposing a task into logical subtasks, where the initial ATN networks encodes the overall structure of the task and the SUBATNs encode specific details to perform a circumscribed component.

Miller has developed a text-generation program, called PROSENET, that uses the traversal of ATN nodes and arcs to control the production of text fragments (Figure 8.2) [84, 85]. PROSENET uses ATN hierarchies to organize the text generation into logical segments, such as sentences, paragraphs, and main topics. This logical decomposition makes text generation with PROSENET easy to adapt to the organizational and rhetorical schemas in TOPAZ. Rennels reimplemented PROSENET using an object-oriented design [102,85]. The TOPAZ text-generation system is based on a simplified version of Rennels' system.

In TOPAZ, ATN nodes are implemented as OZONE objects with three specialized spaces, ENTRY.ACTIONS, ARCS, and EXIT.ACTIONS (Figure 8.3).¹ The elements in the slots ENTRY.ACTIONS and EXIT.ACTIONS are LISP expressions that are always executed whenever an ATN node is traversed. ENTRY.ACTIONS are performed when control first passes to the node; EXIT.ACTIONS are performed just prior to the current node relinquishing control to the next node.

ARC slots are more complicated. ARC slots contain up to four items: (1) a predicate expression, (2) a THEN field, (3) a BEFORE field, and (4) an AFTER field. The BEFORE and AFTER fields are optional; the predicate expression and NEXT fields are required. After all ENTRY.ACTIONS have executed, each predicate in the ARC slot is

¹The ATN routines were written by David Combs, who redesigned and extended a much more limited ATN system that was written by me.

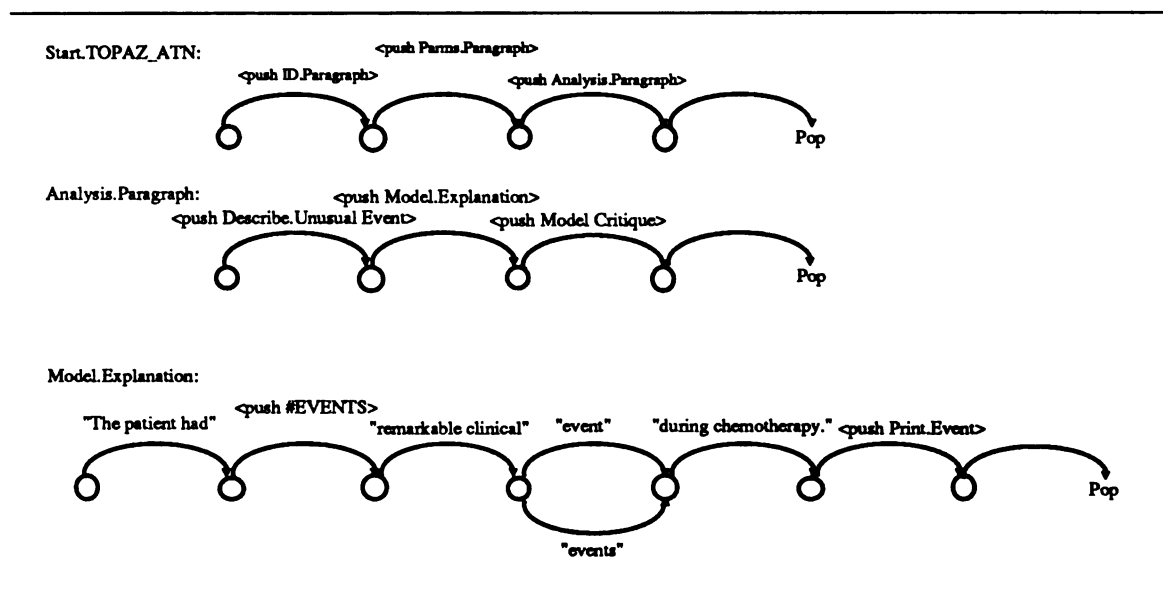


Figure 8.2 The PROSENET methodology for text generation.

PROSENET generates text using hierarchies of augmented transition networks (ATNs). Nodes and arcs generate text fragments. Each node contains logical statements that determines which node to traverse next. Hierarchies of networks organize text generation into progressively larger units—for example, from sentences, to paragraphs, to topics.

evaluated in order. The first arc whose predicate succeeds becomes the *active* arc. At least one arc predicate must succeed, or a system error occurs. The THEN field determines which node gets control after the EXIT.ACTIONS in the current node are executed. The THEN field has two forms: a single name or a three-member list. If a single name appears in the THEN field, control passes to the node with that name within the same ATN hierarchy. If a three-member list appears in the THEN field, the first item is interpreted as the name of the first node in a SUBATN that is to be executed before leaving the current node, the second item is the name of the next node in the current hierarchy that receives control if the SUBATN returns successfully, and the third item is the name of the next node in the current hierarchy that receives control if the SUBATN returns abnormally. When the SUBATN finishes its processing, it returns control back to the calling ATN by having a keyword in the THEN field, rather than a next node name. Four keywords—FINISH, ABORT, SYSTEM-ABORT, and USER-ABORT—return control to the higher network with various condition flags. Finally, an ARC slot may contain BEFORE and AFTER fields, which have lists of functions that are executed before and after an arc is executed. Unlike the ENTRY.ACTIONS and EXIT.ACTIONS, which are always executed whenever a node is traversed, only the BEFORE and AFTER actions on the *active* arc are executed.

8.2 Text-Generation Procedures

The TOPAZ presentation model is composed of two schemas (Section 5.3). The organizational schema organizes a TOPAZ summary into two major topic headings:

1. Patient-Identifying Data
2. Analysis of the Clinical Course

A major heading is constructed from an ATN node that calls a SUBATN to elaborate the key points for each topic. These SUBATNs, in turn, call other SUBATNs to process further specific details relevant to each major topic heading. Arc predicates test for the presence of noteworthy ETNET intervals created by the abstraction task. The existence of specific interval types in a patient's ETNET abstraction forms the basis for generating specific summary statements. If a pertinent ETNET interval is present, the execution of the ATN node generates text that describes that event. Because each patient has a unique

NAME: CRITIQUE.FIT

(ENTRY.ACTIONS

(A1 (STORE.REGISTER (GET.FIT.INTERVALS 'Obs.v.Pred+)))

(A2 (STORE.REGISTER (GET.FIT.INTERVALS 'Obs.v.Pred-)))

(A3 (STORE.REGISTER GET.FIT.POINTS 'Obs.v.Pred+)))

(A4 (STORE.REGISTER (GET.FIT.POINTS 'Obs.v.Pred-))))

(ARC

(A1 (ANY.ENTRY? Obs.v.Pred.Intervals)

THEN CRITIQUE.OBS.V.PRED.INTERVALS)

(A2 (ANY.ENTRY? Obs.v.Pred.Points)

THEN CRITIQUE.OBS.V.PRED.POINTS)

(A3 T

THEN FINISH

BEFORE ((PRINT.FEATURE "There were no periods of
poor model fit during this clinical event."))))

Figure 8.3 A TOPAZ text-generation node.

TOPAZ produces text by traversing a hierarchy of augmented transition networks (ATNs). TOPAZ ATN nodes are constructed from OZONE objects that contain actions and arcs. Actions can store or retrieve values (e.g., STORE.REGISTER), print text (e.g., PRINT.FEATURE), or call another SUBATN. Arcs contain predicates (e.g., ANY.ENTRY?). The first predicate that succeeds is executed. An arc points to the next node to be executed (e.g., CRITIQUE.OBS.V.PRED.INTERVALS), but it may invoke a SUBATN before traversing to the next node (not shown).

set of ETNET intervals based on his clinical findings, each summary has the same overall structure, but has different specific events discussed in each paragraph.

The Patient-Identifying Data paragraph is a stereotypical initial introduction to the patient's clinical problems. Unlike the paragraphs that follow, this segment is easily generated by simple queries into the patient database. The elements in this first paragraph are not generated from ETNET interval abstractions.

The Analysis of the Clinical Course paragraph discusses three main topics. Separate ATNs produce summary text designed for each topic. The first topic is a discussion of the physiological implications of the patient's clinical course. The text generated by the ATN is based on the parameter estimates that were derived when all observations were fitted to the bone-marrow model ($\hat{\phi}$). The ATN for this topic converts the parameter names into their clinical meaning and interprets the parameter values to their clinical implications. For example, the first element in the vector $\hat{\phi}$ is translated to the *bone-marrow mass*, and its value is translated into *normal*, *mildly depressed*, *moderately depressed*, or *markedly depressed*.

The second topic of the Analysis of the Clinical Course paragraph presents the key clinical findings. It is in this part of the summary text that the remarkable clinical events noted by the temporal-abstraction system are presented. For each remarkable clinical event, an ETNODE was created noting the presence of a large difference between the expected and observed clinical course. TOPAZ executes a series of ATN nodes to generate a discussion about the event represented by the ETNODE. These ATN nodes encode the rhetorical schema (Figure 8.1). The rhetorical schema is executed repeatedly for each remarkable event. This ATN calls additional SUBATNs (1) to describe the remarkable clinical event, (2) to present possible causes for the remarkable event, (3) to critique the model's prediction of the event, and (4) to comment on the implications of any episodes of poor model predictions. Each of these features depends on the presence of various ETNODE intervals generated by the temporal-abstraction system. The presence or absence of an interval during the remarkable event triggers or suppresses various text fragments contained in the ATN nodes. For example, an interval of poor model fit is represented as an interval during which a large difference between the patient-specific model predictions and the observed WBC counts was present (Obs.v.Pred intervals). Comments about conclusions that were derived during a period of poor model fit are generated only if an interval of this type exists within the temporal extent of the remarkable clinical event currently being presented in the summary

text.

After all remarkable events have been discussed, the third topic in the Analysis of the Clinical Summary paragraph is a critique of the model fit over the remaining intervals of time that were not contained in any remarkable event. The result is that all episodes of remarkable clinical events and all periods of poor model fit are presented at some point in the clinical summary.

8.3 Problems and Extensions

In Section 5.3, I presented three notions that are especially relevant to clinical summarization:

1. The set of findings and conclusions that have been inferred by the system during the analysis of the clinical record
2. The level of detail desired by the summary reader
3. The body of knowledge that can be assumed by both the reader and the user

The current implementation of the TOPAZ presentation model has focused on addressing only the first point. The current presentation model is insensitive to the characteristics of the clinical reader; it assumes a reader at the level of expertise of a third- or fourth-year medical student. For example, one expert oncologist thought that, for his level of training, the overall level of detail was excessive, although he did not dispute the content of the summaries. What is missing from the TOPAZ presentation model is a user model that can control the level of detail in the summary output according to assumptions about the expertise of the intended reader.

A related shortcoming is the noninteractive nature of the summarization system. A highly desired modification would be a presentation model that would provide a limited amount of information, but then could respond to a user's request for additional details about selected points. Even if a more sophisticated user model were implemented, in the current noninteractive style, a user requesting more information would receive additional details about the entire clinical course rather than just an expansion of only those points of most interest.

The ability to “focus in” on areas of dispute or confusion would greatly enhance the acceptability and usefulness of a system such as TOPAZ. An important feature of the model-based approach in TOPAZ is that there is an intuitive metric that the program could use to determine how much attention it should pay to a particular event; the larger the difference between the expected and predicted clinical course, the more unusual and the more unexpected the event. TOPAZ could generate minimal statements for mildly abnormal events and more extensive statements for more aberrant events. The definition of *minimal* or *extensive* statements could be modified according to the presentation user model.

The TOPAZ presentation model provides a causal explanation of an unusual event based on the relationships embodied in the bone-marrow physiological model. The presentation model has no explicit knowledge about physiological models. The model builder must encode into the presentation model the causal implications of the relationships present in the structural model. A new model, with new features, parameters, and relationships, requires the construction of a new presentation model that can correctly examine the model output.

The interdependency between the physiological model and the presentation model extends to only that portion of TOPAZ that generates a model-based causal explanation of an unusual event. Without any additional knowledge about the underlying structural model, the TOPAZ context model can detect large differences between expected and predicted WBC counts, which enables the presentation model to note regions in the clinical course that are unusual. Thus, in the absence of knowledge about the underlying relationships in the structural model, the system would still be capable of providing useful information about the presence of unusual events in the clinical course, but it would not be able to formulate model-based causal explanations.

8.4 Summary

The text-generation system is the least complex of the three TOPAZ modules. For an expert oncologist, skilled in physiological models, a textual description of the model results may be unnecessary. Even for the physician less familiar with mathematical models, a display of the ETNET abstraction intervals may be sufficient. The TOPAZ presentation model encodes a model of time that differs from those that are found in the process-oriented

bone-marrow model and in the event-oriented interval-abstraction model. The collection of ATN hierarchies that encodes the organizational and rhetorical schema can be viewed as encoding a state-based temporal model. For each abnormal event that can be encoded as an ETNET interval, an ATN node or hierarchy exists in the presentation model to discuss that interval's clinical implications. The presence of a remarkable clinical event means that the patient was in a certain clinical state during that interval of time. A different event implies a different state, which is encoded as a different ATN node or hierarchy in the presentation model and which generates different text descriptions. The TOPAZ presentation model must encode the various clinical states that may be detected during a patient's clinical course. The TOPAZ presentation model also contains a sequential model of time. In general, a coherent clinical presentation does not jump around from point to point in time—rather, it proceeds in a smooth, predictable progression. In the presentation model, each event in time is examined in detail before the next event is presented. The finite structure of ATNs imposes a discrete view of time that is not found in the bone-marrow model; the traversing of linked nodes in ATNs imposes a sequential view of time that is not found in the interval-abstraction model.

The three temporal models in TOPAZ collectively capture a wide range of temporal concepts. The use of multiple temporal models is an explicit acknowledgment of the breadth of the problems present in temporal reasoning. The marked differences in the characteristics of the three models reinforces the view that it is unlikely that a single representational formalism can capture the diversity of the temporal concepts that appear in even relatively simple temporal problems. The development of TOPAZ is an attempt to accommodate the strengths of multiple modeling approaches in a well-coordinated methodology to perform temporal-reasoning and temporal-representation tasks that could not be accomplished by any single modeling formalism alone.

CHAPTER 9

Summary and Conclusions

In this dissertation, I have presented a methodology for reasoning about events that evolve over time. The methodology proposes three steps: (1) model-based transformation of time-ordered observations into underlying concepts, (2) interval-based transformation of model predictions into domain-specific temporal abstractions, and (3) state-based transformation of temporal abstractions into clinically meaningful discourse fragments. In Chapters 6, 7, and 8, I presented the design and implementation details of one approach to realizing these three transformations. For each transformation, I described weaknesses that exist in the current system and I suggested improvements that may address these deficiencies. In the previous chapters, each of the three steps was examined in relative isolation from the other steps. In this chapter, I shall examine the contributions and limitations of this research from a more integrated perspective. The goal of this chapter is to critique the strengths and weaknesses of the TOPAZ methodology and implementation as a conceptual paradigm for representing and reasoning about complex temporal concepts.

9.1 Summary of the Dissertation

TOPAZ is a set of interrelated programs designed to perform a specific task: to summarize time-ordered medical data. TOPAZ embodies several closely related experiments

about temporal reasoning and temporal representation. In Chapter 1, I presented three key hypotheses that are central to this dissertation:

1. The diversity of temporal knowledge that is used in complex temporal problem solving demands multiple representational and inferential formalisms that work cooperatively
2. Explicit structural models that relate time-ordered observations to underlying concepts provide intuitive methods to interpret, explain, and predict subtle temporal patterns
3. Context-specific reasoning limits the size of the problem space by constraining the temporal features and relationships that must be considered in the current reasoning task

The essential features and characteristics of the TOPAZ methodology and implementation reflect these hypotheses. In summarizing the main themes in the dissertation, I shall review the key results that lend support to these underlying hypotheses.

- I have argued that the complexity of temporal knowledge cannot be captured in one formalism that can be used by a computer-based problem solver. For this reason, I have developed a system that manipulates three temporal models—a real-valued continuous-process model, an interval-based discrete-event model, and a state-based discourse model. The discrete-event model is an abstraction of the same underlying reality represented in the continuous-process model, but the type of domain-specific temporal knowledge and concepts encoded in each model differs. The vast differences in the temporal assumptions among the three models constitute the reason why I argue for multiple temporal representations.

Without multiple temporal models, a system such as TOPAZ must encode all domain knowledge using a single modeling formalism. There is no modeling methodology that captures the diversity of temporal facets encoded by the three TOPAZ models. If he was forced to choose only one modeling methodology, the system designer would find that some temporal features currently in TOPAZ could not be represented. TOPAZ needs to analyze and abstract relevant clinical features that have both processlike and eventlike temporal characteristics. Without multiple temporal models, TOPAZ would be limited in the class of temporal features that it could detect and report in a patient summary.

- I have shown how an explicit structural model can be used to derive consistent clinical implications from time-ordered patient data. The relationships encoded in the structural model restrict the range of possible patient-specific estimated parameter values and restrict the implications that can be derived from time-ordered patient data. The TOPAZ physiological model (the bone-marrow model) is essential for TOPAZ to detect subtle physiological implications from patient data that are sparse and error-prone.

I have shown how a structural model can be used to represent different classes of expectations. In TOPAZ, I use the structural model to represent both population-based and patient-specific expectations. By comparing observations and expectations, the system designer can create precise definitions for clinical concepts that previously have been unclear, such as the *expected clinical course* and *unexpected observations*.

I have shown how a structural model that encodes causal relationships can be used to derive causal explanations of an unusual temporal pattern. The structural model parameters and predictions in TOPAZ permit the construction of summary statements that not only *describe* unexpected findings but also *explain* the probable cause of those findings.

- I have shown how TOPAZ can focus and limit symbolic reasoning by using context-specific rules associated with intervals that represent notable events. In TOPAZ, searching for a cause that explains an unusual observation is initiated only if an unusual observation is detected. Similarly, TOPAZ critiques the ability of the structural model to predict the patient's clinical course only if there are intervals during which the model's predictive performance appears to be inadequate. The number of rules in the TOPAZ knowledge base is always larger than the number of rules that need to be examined by the TOPAZ inference engine. As the reasoning context changes and intervals are opened and closed, the set of available rules changes dynamically to reflect new context-specific definitions and new potentially relevant abstractions.

9.2 Limitations of the Methodology

In this dissertation, I have described the use of numeric and symbolic temporal models to interpret and abstract subtle clinical features that evolve over time. By combin-

ing a *processlike*, real-valued temporal model with an *eventlike*, interval temporal model, TOPAZ is able to represent and reason about clinical entities with markedly differing temporal characteristics. The TOPAZ methodology and implementation contain a number of domain assumptions that can severely limit their use in certain tasks.

- The TOPAZ methodology assumes that an accurate structural model of the relationships among observations and underlying processes can be elucidated from experts. Many processes in medicine are not understood with sufficient precision to permit the expert to construct a detailed structural model. Qualitative modeling techniques probably are not the answer in this situation. In data-poor and error-prone settings, qualitative models do not provide sufficient inferencing capabilities to draw conclusions about unobserved system states. In the absence of an accurate structural model, patterns of interest could be encoded as rules in ETNODEs. This approach is equivalent to implicitly encoding a model as rules. In an area where good predictive models can be constructed, I believe encoding a structural model as rules is not the desired approach. In areas where no such structural model exists, ETNODE rules would be the only method available in TOPAZ to represent interesting temporal patterns.
- The TOPAZ temporal models represent temporal sequences poorly. For example, there is no way to represent the phrase: "Patients with this cancer receiving four cycles of treatment *X*, followed by two cycles of treatment *Y*, and, if they show improvement, continue to receive treatment *X* until all evidence of active tumor growth disappears." ETNET could represent the fact that a specific patient received four cycles of treatment *X* followed by two cycles of treatment *em Y* followed by *n* cycles of treatment *em X*, but this would not capture the same temporal information as is contained in the quoted sentence. Musen *et al.* [92,91] describe a graphical language for specifying procedural knowledge in cancer-chemotherapy protocols. Their techniques could encode the example sequence information. TOPAZ could make use of this additional sequence knowledge to detect deviations from the expected therapeutic course. Without this temporal knowledge, TOPAZ only notes whether the patient observations are unusual, given the therapy that the patient actually received.
- The TOPAZ temporal models also represent temporal uncertainty poorly. The physiological model requires that the times of observations and therapies be known accu-

rately. The TOPAZ bone-marrow model represents the presence of random error in patient observations, but it does not represent random error in the input (therapy) or temporal uncertainty. Statistical techniques that incorporate these additional sources of uncertainty exist, but, in the data-poor TOPAZ environment, they are unlikely to be helpful. The TOPAZ interval model (ETNET) has no capability of representing temporal imprecision or uncertainty among events. Imprecision in the starting and ending times for events could be added by extending the definition of ETNET event-times to include a variance (or fuzz) term. There is no simple extension that would support uncertain temporal relationships among events. Allen's interval calculus (Section 2.1.1) can model uncertain temporal relationships. In a domain where uncertain temporal relationships were important, I would add a model based on Allen's calculus. This type of temporal model was not required in the summarization task performed by TOPAZ.

In the following sections, I shall use the three research hypotheses to examine the relationship of the new contributions embodied in this work to previous works in temporal reasoning and representation.

9.3 Multiple Temporal Formalisms

TOPAZ relies on the expressiveness of three distinct models to capture the breadth of temporal reasoning required to summarize time-ordered clinical data. Each model uses a different formalism to encode different temporal concepts necessary to solve one particular temporal task. The bone-marrow model uses a system of differential equations to model the physiological processes that relate changes in peripheral WBC counts to the underlying myelosuppressive effects of cancer chemotherapy. The context model uses a collection of symbolic intervals to represent aggregates and abstractions of events that have both clinical meaning and temporal duration. The presentation model uses a state-based transition network to model the medical rhetorical style of describing clinical events that evolve over time. The three TOPAZ models differ greatly in the structure of time they assume, the temporal concepts they encode, and the temporal tasks they perform. Collectively, the models capture a broad range of temporal facets that would be impossible to represent using one of the three model formalisms alone.

Researchers in knowledge representation attempt to describe key features of the real world in a format that allows an automated reasoner to solve difficult problems. The key research issues in knowledge representation are (1) finding a formalism that captures the essential elements and relationships in the real world, and (2) defining a reasoning procedure that manipulates the formalism to solve useful problems. A major tenet of knowledge-representation research is that a formalism with features that are isomorphic to the key structures in the problem domain facilitates mapping concepts between the model and reality. The construction of a single model formalism to encode all requisite knowledge about a problem domain is desirable for a number of reasons, such as simplifying knowledge engineering and knowledge-base maintenance. In temporal-representation research, the works of Allen, Ladkin, Vilain, McDermott, and others (Section 2.1.1) seek to define uniform logic-based languages for expressing temporal knowledge. The work of Kohane with TUP and THRIPT (Section 2.2.1.2.2) is an application of Allen's notion of time as sets of constraints between temporal intervals as the uniform temporal formalism to reason about a patient's past medical history. In rule-based expert systems, the work of Fagan in VM (Section 2.3.3.1) used production rules with temporal predicates as the uniform temporal formalism to represent state-based changes over time (Section 2.3.3.1) and thus to reason about ventilator-management changes in postsurgical patients. In database systems, the works of Ariav, Clifford, and Snodgrass model time as changes in database tuples or in tuple attributes (Section 2.2). The unifying theme of these vastly different approaches to temporal knowledge representation (constraint networks, production rules, and databases) is the emphasis on a single model of time to capture all salient features of the problem domain.

TOPAZ endorses multiple temporal representations. In this setting, each representation models specific temporal features well, but is limited in the variety of temporal facets that it can capture alone. This perspective is analogous to Chandrasekaren's model of reasoning experts who share their unique but limited expertise to solve difficult problems [22,21]. As they are in TOPAZ, multiple symbolic temporal representations were the hallmark of Kenneth Kahn's TIME SPECIALIST and Mittal's PATREC (Section 2.2.1.2.1). The TIME SPECIALIST used three symbolic structures—date lines, reference events, and before-after chains—to represent imprecise and uncertain temporal relationships. PATREC contained similar representation structures, and also supported temporal abstraction by clustering related events into a single *key event*. Unlike TOPAZ, neither the TIME SPE-

CIALIST nor PATREC represented causal relationships between stored events.

The combination of numeric causal models and symbolic temporal models was used in the DIGITALIS THERAPY ADVISOR (Section 2.3.4.1) and in the HEART FAILURE PROJECT (Section 2.3.4.2). The DIGITALIS THERAPY ADVISOR combined a numeric model of digitalis kinetics that calculated estimated blood digitalis levels with a symbolic model of digitalis toxicity that adjusted model parameter values based on key clinical findings. The HEART FAILURE PROJECT (HFP) is an ongoing research program that implements a causal hemodynamic model of the cardiovascular system using a network symbolic model. Numerical methods solve differential equations within network nodes. Given a patient state, HFP is able to generate accurate hemodynamic predictions. Unlike TOPAZ, HFP can neither take primary observations as input and generate the most likely hemodynamic explanation, nor derive temporal abstractions about the underlying system from a series of time-ordered observations. Time-sensitive database-access and statistical-regression methods were combined in Blum's RX. The access methods and regression equations in RX can be considered as two distinct temporal models. RX used causal medical knowledge to eliminate regression models that were physiologically implausible. Causal knowledge was also used to constrain the temporal queries into the RX database such that all queries *made medical sense*.

The temporal models in TOPAZ acknowledge two central temporal perspectives: (1) time as a continuous flow, and (2) time as a series of discernible events. The bone-marrow model represents time as an independent real-valued variable. In this model, time is infinitely continuous and maximally dense. The context model represents time as chunks or blocks. Unlike continuous time, events are discrete and therefore enumerable. If we assume that there exists only one underlying reality being represented by these two temporal perspectives, then what is the relationship between the bone-marrow model and the context model? There are two possibilities to consider: (1) the two models represent alternative or competing perspectives of the same temporal phenomenon, or (2) the two models are related conceptually to each other, but differ in their temporal characteristics. For the two models in TOPAZ, the second possibility seems to hold. The most detailed view of time is as a real-valued variable. When we define periods of time as events, continuous time becomes discretized into meaningful units. Because the perspective of *time as events* is an abstraction of *time as flows*, the interval-based context model is an abstraction or simplification of the process-based bone-marrow model. Figure 9.1 depicts a spectrum of

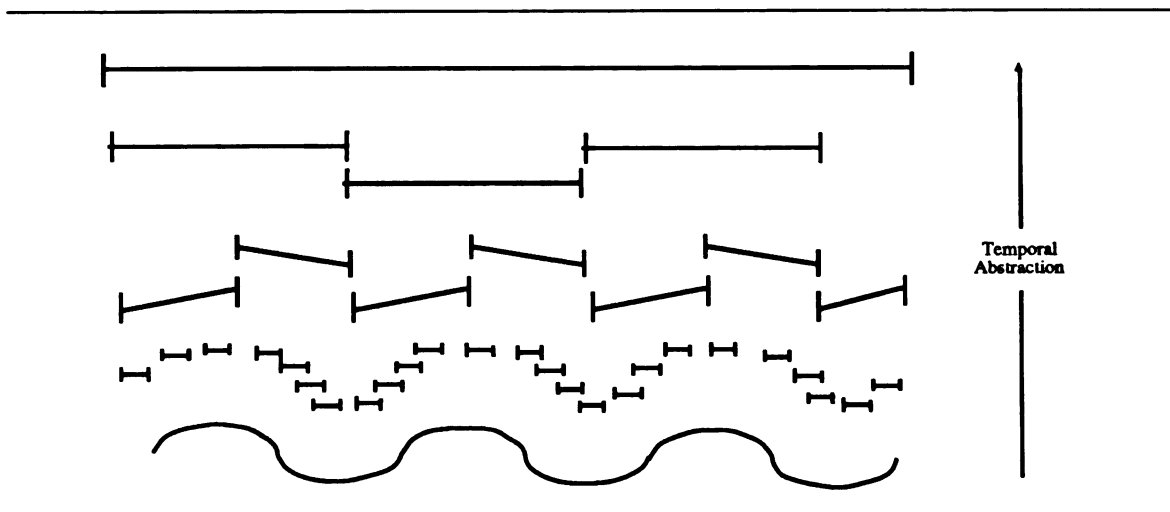


Figure 9.1 Continuous and discrete temporal models.

Time is represented in most detail as a real-valued entity. It is most closely modeled using real-valued models. By discretizing time, we can create temporal abstractions. Very small temporal segments can approximate the continuous model; larger temporal segments are interval abstractions of the underlying real-valued time. The bone-marrow model is a real-valued model; the context model is a discrete model.

temporal abstractions, starting with the continuous, real-valued model, proceeding through small, local discrete approximations to more global discrete approximations, and ending with a single, all-encompassing atemporal abstraction. A continuous model may be approximated by a discrete model. For example, the equations that comprise the bone-marrow model (Figure 6.2) are approximated using a Runge–Kutta numeric integrator, which finds solutions to continuous differential equations by segmenting time into small, discrete linear steps.

Once the system designer has made the decision to support more than one temporal-modeling technique in an expert system, he eventually must face three new issues:

1. If there are multiple model formalisms from which to choose for inclusion into an expert system, how does the system designer know when a method is useful in his application and when a method does not add sufficient expressive power to warrant its inclusion? Answering this question requires some metric for evaluating the usefulness of a candidate representation method. The expressive power of a knowledge-representation formalism depends on both the relevant concepts that need to be en-

coded and the reasoning tasks that need to be performed. Therefore, a meaningful evaluation metric is both problem- and task-specific.

In temporal reasoning, a careful analysis of the temporal characteristics of the problem task can yield important features that can be used to evaluate the usefulness of a specific model formalism. For example, the usefulness of a specific temporal representation usually depends on the temporal granularity required by the problem-solving task. The term *temporal granularity* refers to the smallest or usual temporal interval represented in and reasoned by the problem solver. A good match in temporal granularity can improve the ease in which domain-specific concepts can be encoded in the knowledge base. The key point is that the system designer must understand the temporal characteristics of the problem domain and of the modeling formalism to determine whether the additional representational and inferential capabilities will be useful.

TOPAZ does not offer a methodology for evaluating the usefulness of competing representation formalisms. In the initial phases of designing TOPAZ, I considered nonnumeric methods such as Kuipers' qualitative models (Section 2.1.2) for the bone-marrow model, and ATN networks similar to those used in OBOE (Section 3.4.1) for the context model. In both cases, I discarded these alternative representational methods because they lacked expressive power. For example, current qualitative models do not have sufficiently powerful fitting procedures to generate the most plausible parameter values from a sparse data set. An augmented transition network is a poor representation technique for encoding context models because of the rapid proliferation of ATN nodes for every combination of possible concurrent contexts (Section 3.4.1). These representation techniques did not support the temporal granularity needed to solve the temporal task.

2. If there are multiple ways to represent the same concept, should the concept be encoded redundantly within each formalism? Avoiding redundancy is usually desired for system construction, verification, and maintenance reasons. Humans have much redundancy in both knowledge content and knowledge access. Redundancy might improve expert performance by making knowledge available in settings in which it normally would not be considered. In a system that could express the same concept in multiple formalisms, can the tradeoff between redundancy and no redundancy be

evaluated quantitatively from both a system-performance and a system-maintenance perspective?

3. In a system that supports multiple levels of temporal abstraction, how does the knowledge engineer decide at which level a concept should be encoded? In certain situations, some characteristic of the concept makes one level preferred over the other. In TOPAZ, *processlike* knowledge fits well into the physiological model, whereas *eventlike* knowledge fits well into the context model. In other settings, the intended use of the knowledge may provide some clue about which abstraction level best matches a specific concept. Current practice is to encode a concept at whichever abstraction level seems to fit best. This decision is based on intuition and on past experience.

The marked difference in the temporal characteristics of the modeling techniques in TOPAZ eliminates the problem of selecting the best formalism to encode a specific concept. Concepts that require a continuous *process* view of time are encoded in the physiological model; concepts that require a discrete *event* view of time are encoded in the context model. Because all required concepts readily fit into only one of the two models, TOPAZ also avoids the problem of redundancy in the knowledge base.

The current state of the art in knowledge representation does not provide guidelines for selecting and using multiple representations in expert systems. The issue of knowledge redundancy is also open to active debate. If TOPAZ had more levels of temporal abstraction with less distinction in the type of temporal features encoded at each level, both of these fundamental knowledge-representation issues would have appeared. The choice of the appropriate level of abstraction to represent and reason about events will be dictated by the level of detail required by the reasoning process. As more levels of abstraction become available, the knowledge engineer must decide which level of abstraction best fits each domain-specific concept.

9.4 Model-Based Data Interpretation

A central assumption in TOPAZ is that readily available time-ordered clinical measurements have important decision-making implications an intelligent agent cannot infer by inspecting single observations. TOPAZ uses an explicit model of the dynamic relationships between observed measurements and unobserved physiological entities to relate time-ordered

observations to implications. The use of well-characterized statistical techniques (weighted least-squares fitting) results in the *most likely* patient-specific model from a particular clinical record. By inspecting predictions generated by this model, TOPAZ formulates temporal abstractions to produce summary statements that have more clinical relevance than would a simple description of the data.

In TOPAZ, a detailed causal model is essential to the temporal-abstraction process. The physiological model embodies a set of physical constraints that limits the possible causal explanations for a set of time-ordered patient observations. The model generates predictions for both an observed patient feature (the peripheral WBC count) and unobserved physiological states (the bone-marrow steady-state level, the sensitivity to chemotherapy drugs). Using a time-ordered series of model predictions, the symbolic context model derives temporal abstractions that represent unusual clinical events and their likely physiological cause. The constraints enforced by the bone-marrow model ensure that the conclusions derived by the context model are both physiologically reasonable and clinically relevant. In addition, the logical consistency ensured by the physiological model greatly simplifies the complexity of the rule predicates required in the TOPAZ context model. For example, the physiological model combines multiple courses of chemotherapy with overlapping durations in a consistent manner. The rule predicates that attempt to determine whether excessive drug effect is the cause of an episode of myelosuppression do not need to calculate the cumulative effects of all previous courses of therapy to examine this possibility.

The parameterized causal model enables TOPAZ to have precise definitions for vague concepts such as *the normal clinical course*, *an unexpected observation*, and *a likely physiological cause*. For each patient's unique sequence and dosage of chemotherapy, model predictions using population-based model parameters (ϕ_0) define the typical clinical course, the expected observations, and the normal physiological states. TOPAZ compares the model predictions using patient-specific model parameter estimates ($\hat{\phi}$) to detect unusual events and states. Previous programs have used other methods to encode the expected values of clinical attributes to detect large deviations from the expected clinical course. VM explicitly encoded the normal expected values in tables that were indexed by discrete states. When a patient moved from one state to another, VM replaced the previous expected values with the new expected values for the new state. Unlike TOPAZ, VM did not contain a physiological model, did not generate predictions about unobserved underlying physiological states, and did not represent time as a real-valued quantity. TOPAZ would resemble VM if the interval-

based context model were to derive temporal abstractions using static tables rather than population-based model predictions. Even so, it is not clear that TOPAZ could successfully generate correct patient-specific abstractions with simple state-specific tables of expected values. Because each patient can potentially receive slightly different drug dosages at times different from all others, each patient has unique expected values that change each day. It would be difficult to encode the expected values for all potential schedules and dosages of chemotherapy for each treatment day in table format. Also, without an underlying causal model, a table-based version of TOPAZ would not be able to estimate a probable cause for the deviant observations. VM generated therapeutic conclusions from state-specific expectations and current patient observations. Because there was no physiological model of ventilation and respiration similar to the bone-marrow model in TOPAZ, VM did not derive therapeutic conclusions from a patient-specific model and underlying physiological states.

Unlike VM and TOPAZ, Downs' summarization program (Section 3.3.3) took the opposite approach to representing time-changing expectations by explicitly encoding the set of *abnormal* observations that must exist for the program to conclude an unusual observation or state. The presence of specific abnormal findings in the proper temporal sequence was used as evidence for the existence of an abnormal clinical state. Downs used a probability-based updating scheme to combine evidence for and against a potential abnormal clinical state. No underlying physiology was explicitly encoded in his system, although the expert clinician who defined the temporal pattern-matching rules clearly used such knowledge. Downs' temporal model was discrete and discontinuous; the existence of an abnormal state was determined separately for each visit. Contiguous visits were used only to provide confirmatory or disconfirmatory evidence for the existence of an abnormal state on a specific visit. The process began anew for the next patient visit. No concept of a process that begins during the patient's clinical course, continues for a number of visits, and finally terminates could be encoded in the Downs' discrete temporal model. Downs recognized this shortcoming because it also compromised the assumptions of his probability-based updating scheme, which required conditional independence between adjoining visits. If a patient has an abnormal clinical state on one visit, he is much more likely still to have that abnormal state on his next visit—this increased likelihood of increased abnormal states between adjacent visits violates the conditional independence of adjacent visits. In TOPAZ, the dependencies between visits are explicitly encoded in the physiological model. Uncertainty in TOPAZ is

represented in the covariances of the estimated patient-specific model parameters and in the covariances of the model predictions. This approach to uncertainty removes the problem of conditional dependencies. Although the current implementation of TOPAZ does not use the uncertainty of the parameters and predictions to modify its abstractions or summary statements, the covariance information is calculated and is available. Only additional context rules and a more elaborate discourse network would need to be added to the current system for an analysis of the estimation uncertainties to appear in the final summary text.

The bone-marrow model in TOPAZ is analogous to the parameterized model of digitalis pharmacokinetics in the DIGITALIS THERAPY ADVISOR (Section 2.3.4.1). The DIGITALIS THERAPY ADVISOR used symbolic logic to modify model parameter values to create a patient-specific model of digitalis metabolism. Based on these patient-specific model parameter values and pharmacokinetic model predictions, the program proposed changes in digitalis therapy. Unlike the DIGITALIS THERAPY ADVISOR, TOPAZ uses statistical rather than heuristic techniques to modify population-based model parameters into patient-specific values. TOPAZ uses model predictions to examine and explain past clinical events, whereas the DIGITALIS THERAPY ADVISOR used model predictions to examine and modify future drug levels. Unlike TOPAZ, the DIGITALIS THERAPY ADVISOR did not incorporate multiple temporal models. Without the temporal-abstraction features provided by ETNET, the DIGITALIS THERAPY ADVISOR was not able to predict how long and over what time interval a patient would have toxic or subtherapeutic digitalis levels. Because digitalis metabolism is altered by many frequently used drugs, ETNET also would be a useful addition to the DIGITALIS THERAPY ADVISOR for detecting periods of overlapping administration of potentially interacting drugs. In its current form, the DIGITALIS THERAPY ADVISOR merely can detect the presence or absence of an interfering agent on the current visit, and can alter the current model parameters accordingly. If a therapeutic agent was discontinued recently and that drug has persisting effects on digitalis metabolism (phenobarbital, an inducer of digitalis metabolism, has this behavior), the DIGITALIS THERAPY ADVISOR would recognize only that the interacting drug currently is not being given, and the program would incorrectly assume that no change in model parameters was needed. In TOPAZ, a physiological model of the drug interaction would capture the expected change in digitalis kinetics, and the creation of an ETNET interval would represent the presence of an interacting drug effect and altered drug metabolism. A reasoning system could test for the existence of this ETNET interval to

eliminate the temporal-reasoning error likely to be made by the DIGITALIS THERAPY ADVISOR.

9.5 Context-Sensitive Reasoning

In complex domains, the amount of potentially relevant knowledge needed to solve a useful set of difficult problems can quickly become enormous. To create an inferencing system that provides acceptable performance, expert-system designers frequently must face the problem of controlling the size of the inferencing task. The most commonly used technique is to limit the knowledge that is available to the reasoning system, which reduces the number of possible inferences that can be made by the expert system. Limiting the available knowledge is done by partitioning the knowledge base. The system designer must decide how to structure the knowledge base so that the relevant portions are available for the typical problems while the remainder of the knowledge is accessible for the unusual situation.

The problem-solving context provides a useful guideline for determining what portion of the knowledge base is likely to be useful for solving a particular task. In temporal domains, the problem-solving context is changing constantly. This dynamic environment makes the usual methods for partitioning the knowledge base more difficult to implement. In medical settings, clinical events, such as pathological findings and therapeutic responses, begin and end throughout a patient's illness. Decomposing the reasoning task into logical units is essential for effective medical problem solving. In the presence of temporal changes, the logical units that apply at one moment in the clinical course may not be relevant to the reasoning tasks at another instant. In TOPAZ, ETNET is used to partition the available knowledge in a constantly changing context. With ETNET, rules are associated with a specific clinical event type. These rules are made available to the inferencing system only when the clinical event is present. As new events begin and old events end, rules associated with each event type are added or removed from the set of rules available to the reasoning task. The detection of atypical events evokes the creation of an ETNET interval that contains additional rules applicable to problem solving in that unusual context. For the knowledge engineer, only those rules that apply over the duration of a very specific context need be encoded. Because rules within instantiated contexts are permitted to define the existence of related clinical contexts, contextual dependencies between clinical events and abstractions

are easy to encode and maintain.

Zisman [132] described a methodology for the symbolic modeling of asynchronous, concurrent processes that combined Petri nets and rules. Nodes and arcs in a Petri net provided a method for encoding relationships among interacting or independent processes; rules associated with nodes provided a method for encoding process-specific knowledge. This representation is analogous to using ETNODEs to encode clinical contexts and rules within ETNODEs to encode context-specific knowledge. By permitting more than one node in the Petri net to be “activated,” Zisman could model the effects of multiple concurrent processes. This feature of Petri nets is identical to the use of multiple “open” ETNODEs to model the presence of multiple concurrent clinical contexts. Nodes in Zisman’s Petri net represented instantaneous changes in process, whereas nodes in ETNET represent intervals that have temporal duration. This semantic difference reflects the different temporal models encoded by the two methods: Zisman’s models are state-based transition models, whereas ETNET models are interval-based event models.

Unlike TOPAZ, which adds and removes rules as the problem context changes, VM changed the definition of symbolic values and kept the basic rule set unchanged. For example, in VM, the range of measurement values that mapped to the symbolic value *high* for systolic blood pressure (SBP) would change when the patient changed ventilator modes. In TOPAZ, this same behavior would be accomplished by having multiple rules for concluding *high* SBP—one rule associated with each clinical context (ventilator mode). When the patient changed ventilator modes, the ETNET interval denoting the termination of the previous ventilator mode would be closed, removing its rule for SBP, and the ETNET interval denoting the new ventilator mode would be instantiated, adding its rule for SBP into the set of available rules. If a patient went from respirator mode A to mode B and then back to mode A, VM had the ability to use different symbolic definitions for each instance of mode-A treatment. With ETNET, TOPAZ would accomplish this sequence-sensitive behavior by defining event types that distinguish between the first and subsequent instances of mode-A treatment and therefore to reason differently about the first and later treatments. This would be an important distinction if the first instance represented the normal progression of therapy but later instances represented a return to earlier treatment due to the patient’s inability to tolerate the next treatment mode (for example, the inability to tolerate mode B in the A–B–A example).

ETNET can be viewed as a frame-based representation system, where a frame

denotes an interval of time and each instance contains frame-specific rules. From this perspective, ETNET is akin to the “activation” net in the PRESENT ILLNESS PROGRAM (PIP) [99]. In PIP, the detection of a clinical finding would “activate” a node within the knowledge base that contained parameters and rules that applied to the context represented by that node. In addition, the activation of a node would also partially activate nodes that represented clinical events that either were closely related to the active node or were usually seen in conjunction with the clinical event. This behavior is similar to the addition of rules in ETNET that may instantiate other ETNET intervals. The set of new ETNET intervals (PIP = concepts) that may be created (PIP = detected) is dependent on the set of ETNET intervals (concepts) currently open (PIP = active). Unlike ETNET intervals, PIP concepts did not denote temporal entities, and PIP did not perform significant temporal reasoning.

ETNET provides a flexible representation and reasoning environment for encoding and partitioning context-dependent knowledge. In TOPAZ, I have focused on time as the source of context-sensitive knowledge. ETNET does not assume any special features of time or medicine that would limit the semantics of ETNET objects solely to representing clinical temporal intervals.

9.6 Looking Forward

Throughout this dissertation, I have focused on understanding the temporal evolution of past patient observations. Langlotz *et al.* [71] presented the design of a decision-support program that would use patient-specific predictions to assess the likely future outcome of alternative treatment plans. Also included in their design was a utility model that would help to choose the best overall plan among competing treatment plans. Without significant modifications, TOPAZ could perform the patient-specific prediction task and aid in the utility-analysis task for a therapy planner based on these principles. The patient-specific bone-marrow model can predict the expected physiological effects of a simulated treatment plan at any future time. ETNET could generate intervals denoting periods of unacceptable toxicity or undertreatment. With the predictions and abstractions generated by TOPAZ, a physician could understand the likely implications of selecting a specific treatment plan.

Models make predictions about the real world. When a prediction about a future event is not realized, something unexpected or unknown has happened. This failure to observe a prediction is a clear signal that the predictor must do additional reasoning to

understand why the real world did not follow the modeled world. The list of possible causes for a failed prediction is large; determining a specific cause for a failed prediction may be extremely difficult. The main point for computer-based reasoning systems is that large discrepancies between observations and predictions provide an easily recognized warning that the internal representation of the world may not be in agreement with the external, real world, and that something needs to be done about this possible mismatch. In TOPAZ, I examined differences between patient-specific model predictions and patient observations to critique the degree of belief in an inferred abstraction generated by the system. This idea readily extends to changing current and future patient-specific clinical predictions based on unrealized patient-specific model predictions. For example, assume that the patient-specific model predicts steady improvement with current treatment, but that subsequent clinical observations show that the patient is not improving as anticipated. At some point, a reasoning system could decide that the differences between the model predictions and patient observations are too large to ignore, and could initiate a set of reasoning tasks to determine why the predictions and measurements disagree.

Expert systems have been criticized for being unable to detect when they have incorrect or insufficient knowledge about the problem domain. A decision-support program that had the capability to monitor differences between internal model predictions and real-world observations and to trigger additional reasoning resources when it notes large differences, could detect when it is in trouble, when its internal representation is incorrect, and when its inferences should be suspect. TOPAZ contains sufficient information about parameter estimates and model predictions to provide a statistically sound definition of a *large difference*. This additional capability should improve the robustness and acceptability of these systems.

9.7 Summary

TOPAZ is a collection of techniques and approaches, each with specific representational and inferencing strengths and weaknesses, that work together to solve a class of temporal problems. Each task in TOPAZ has a unique temporal model that fits the temporal features of the problem that this task is designed to solve. The use of specific temporal models for specific temporal tasks is a “piecewise solution” to temporal reasoning. There are temporal-reasoning situations that TOPAZ does not handle, such as solving problems

where there is temporal uncertainty in the observations. I believe future research in knowledge representation will create new approaches to specific temporal problems, but that no single method will solve “the time problem.” My design of TOPAZ reflects this belief.

Bibliography

- [1] I Ahn and R Snodgrass. Performance evaluation of a temporal database management system. In C Zaniolo, Editor, *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 96–107, Washington DC, 1986.
- [2] DM Allen. Parameter estimation for nonlinear models with emphasis on compartmental models. *Biometrics*, 39:629–637, 1983.
- [3] JF Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.
- [4] JF Allen. Towards a general theory of actions and time. *Artificial Intelligence*, 23:123–154, 1984.
- [5] JF Allen and PJ Hayes. A common-sense theory of time. In *Proceedings IJCAI-85*, pages 528–531, Los Angeles, CA, 1985. Morgan Kaufmann.
- [6] JF Allen and PJ Hayes. Moments and points in an interval-based temporal logic. Technical Report TR-180, Department of Computer Science, University of Rochester, Rochester, NY, 1987.
- [7] JF Allen and HA Kautz. A model of naive temporal reasoning. In JR Hobbs and RC Moore, Editors, *Formal Theories of the Commonsense World*, Chapter 7, pages 251–268. Ablex, Norwood, NJ, 1986.
- [8] G Ariav. Preserving the time dimension in information systems. PhD thesis, Department of Decision Sciences, University of Pennsylvania, Philadelphia, PA, 1983.

- [9] A Barr and EA Feigenbaum. *The Handbook of Artificial Intelligence*. Morgan Kaufmann, Los Altos, CA, 1981.
- [10] JR Beck and SG Pauker. The Markov process in medical prognosis. *Medical Decision Making*, 3:419–458, 1983.
- [11] MB Bischoff, EH Shortliffe, AC Scott, RW Carlson, and CD Jacobs. Integration of a computer-based consultant into the clinical setting. In RE Dayhoff, Editor, *Proceedings Symposium on Computer Applications in Medical Care*, pages 149–152, Baltimore, MD, 1983. IEEE Computer Society.
- [12] BI Blum. *Information Systems for Patient Care*. Springer-Verlag, New York, NY, 1984.
- [13] BI Blum. *Clinical Information Systems*. Springer-Verlag, New York, NY, 1986.
- [14] RL Blum. Displaying clinical data from a time-oriented database. *Computers in Biology and Medicine*, 11:197–210, 1981.
- [15] RL Blum. Discovery and representation of causal relationships from a large time-oriented clinical database: The RX project. In DA Lindberg and PL Reichertz, Editors, *Lecture Notes in Medical Informatics*, Volume 19. Springer-Verlag, New York, NY, 1982.
- [16] DG Bobrow. *Qualitative Reasoning about Physical Systems*. MIT Press, Cambridge, MA, 1985.
- [17] A Bolour, TL Anderson, LJ Dekeyser, and H Wong. The role of time in information processing: A survey. *ACM SIGART Newsletter*, 80:28–48, 1982.
- [18] B Breutmann, EF Falkenberg, and R Mauer. CSL: A language of defining conceptual schemas. In G Bracchi and GM Nijssen, Editors, *Proceedings IFIP Working Conference on Data Base Architecture*. North Holland, Amsterdam, 1979.
- [19] RF Brown. Compartment system analysis: State of the art. *IEEE Transactions Biomedical Engineering*, BME-27:1–11, 1980.
- [20] BC Bruce. A model for temporal references and its application in a question answering program. *Artificial Intelligence*, 3:1–25, 1972.

- [21] B Chandrasekaran. Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. *IEEE Expert*, 1:23–30, 1986.
- [22] B Chandrasekaran. Towards a taxonomy of problem solving types. *AI Magazine*, 4:9–17, Winter/Spring 1983.
- [23] B Chandrasekaran and S Mittal. Conceptual representation of medical knowledge for diagnosis by computer: MDX and related systems. In M Yovits, Editor, *Advances in Computers*, pages 217–293. Academic Press, New York, NY, 1983.
- [24] J Clifford. Formal semantics for time in databases. *ACM Transactions on Database Systems*, 8:214–254, 1983.
- [25] J Clifford. The Historical Relational Data Model (HRDM) and algebra based on lifespans. In *Proceedings Third International Conference on Data Engineering*, pages 528–537, Los Angeles, CA, 1987. IEEE Computer Society.
- [26] J Clifford and AU Tansel. On an algebra for historical relational databases: Two views. In S Navathe, Editor, *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 247–265, Austin, TX, 1985. ACM Press.
- [27] GF Cooper. A diagnostic method that uses causal knowledge and linear programming in the application of Bayes' formula. *Computer Methods and Programs in Biomedicine*, 22:223–237, 1986.
- [28] BJ Cox. *Object-Oriented Programming: An Evolutionary Approach*. Addison-Wesley, Reading, MA, 1986.
- [29] B Dawson-Saunders, TA Mast, WT Finch, HR Konrad, and JR Folse. Content knowledge and problem-solving skill in reviewing medical charts. *Medical Education*, 18:31–35, 1984.
- [30] IM de Zegher-Geets. IDEFIX: Intelligent summarization of a time-oriented medical database. Master's thesis, Medical Computer Science Group, Stanford University, Stanford, CA, 1987.
- [31] IM De Zegher-Geets, AG Freeman, MG Walker, RL Blum, and G Wiederhold. Summarization and display of on-line medical records. *MD Computing*, 5:38–45, 1988.

- [32] TL Dean. Time map maintenance. Technical Report YALEU/CSD/RR-289, Department of Computer Science, Yale University, New Haven, CT, 1983.
- [33] TL Dean. Temporal imagery: An approach to reasoning about time for planning and problem solving. Technical Report YALEU/CSD/RR-433, Department of Computer Science, Yale University, New Haven, CT, 1985.
- [34] TL Dean and DV McDermott. Temporal data base management. *Artificial Intelligence*, 32:1-55, 1987.
- [35] KG Denbigh. *Three Concepts of Time*. Springer-Verlag, Berlin, 1981.
- [36] VT DeVita, S Hellman, and SA Rosenberg. *Cancer, Principles and Practice of Oncology*, 2nd edition. Lippincott, Philadelphia, PA, 1985.
- [37] SM Downs. A program for automated summarization of on-line medical records. Master's thesis, Medical Computer Science, Stanford University, Stanford, CA, 1986.
- [38] SM Downs, MG Walker, and RL Blum. Automated summarization of on-line medical records. In R Salamon, B Blum, and M Jorgensen, Editors, *MEDINFO 86*, pages 800-804, Washington DC, 1986. North-Holland.
- [39] LM Fagan. VM: Representing time-dependent relations in a medical setting. PhD thesis, Department of Computer Science, Stanford University, Stanford, CA, 1980.
- [40] LM Fagan, JC Kunz, EA Feigenbaum, and JJ Osborn. Extensions to the rule-based formalism for a monitoring task. In BG Buchanan and EH Shortliffe, Editors, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Chapter 22, pages 397-423. Addison-Wesley, Reading, MA, 1984.
- [41] KD Forbus. Interpreting observations of physical systems. Technical Report UIUCDCS-R-86-1248, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 1986.
- [42] KD Forbus. The qualitative process engine. Technical Report UIUCDCS-R-1288, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 1986.
- [43] P Fraisse. *The Psychology of Time*. Harper & Row, New York, NY, 1963.

- [44] JF Fries and DJ McShane. ARAMIS (The American Rheumatism Association Medical Information System) — A prototypical national chronic-disease data bank. *Western Journal of Medicine*, 145:798–804, 1986.
- [45] M Gibaldi and D Perrier. *Pharmacokinetics*. Marcel Dekker, New York, NY, 1982.
- [46] AA Goetz, MJ Peters, R Folse, et al. Chart review skills: A dimension of clinical competence. *Journal of Medical Education*, 54:788–796, 1979.
- [47] PB Gove. *Webster's Third New International Dictionary of the English Language, Unabridged*. G. and C. Merriam, Springfield, MA, 1971.
- [48] PJ Hayes. What the frame problem is and isn't. In ZW Pylyshyn, Editor, *The Robot's Dilemma: The Frame Problem in Artificial Intelligence*, Chapter 7, pages 123–138. Ablex, Norwood, NJ, 1987.
- [49] PJ Hayes and JF Allen. Short time periods. In *Proceedings IJCAI-87*, pages 981–983, Milan, Italy, 1987. Morgan Kaufmann.
- [50] DH Hickam, EH Shortliffe, MB Bischoff, AC Scott, and CD Jacobs. The treatment advice of a computer-based cancer chemotherapy protocol advisor. *Annals of Internal Medicine*, 103:928–936, 1985.
- [51] JR Hobbs and RC Moore. *Formal Theories of the Commonsense World*. Ablex, Norwood, NJ, 1985.
- [52] JC Hogge. TIME and TPLAN user's manual. Technical Report UIUCDCS-R-87-1366, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 1987.
- [53] JC Hogge. TPLAN: A temporal interval-based planner with novel extensions. Technical Report UIUCDCS-R-87-1367, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 1987.
- [54] PJ Hopper, Editor. *Symposium on Tense and Aspect*, Amsterdam, 1982. J. Benjamin.
- [55] KM Kahn. Mechanization of temporal knowledge. PhD thesis, Project MAC, Massachusetts Institute of Technology, Cambridge, MA, 1975.

- [56] KM Kahn and GA Gorry. Mechanizing temporal knowledge. *Artificial Intelligence*, 9:87–108, 1977.
- [57] MG Kahn, JC Ferguson, EH Shortliffe, and LM Fagan. Representation and use of temporal information in ONCOCIN. In MJ Ackerman, Editor, *Proceedings Symposium on Computer Applications in Medical Care*, pages 172–176, Baltimore, MD, 1985. IEEE Computer Society.
- [58] MR Klopprogge. TERM: An approach to include the time dimension in the entity-relationship mode. In PPS Chen, Editor, *Entity-Relationship Approach to Information Modeling and Analysis*, pages 477–512. ER Institute, Saugus, CA, 1981.
- [59] IS Kohane. Temporal reasoning in medical expert systems. In R Salamon, B Blum, and M Jorgensen, Editors, *MEDINFO 86*, pages 170–174, Washington DC, 1986. North-Holland.
- [60] IS Kohane. Temporal reasoning in medical expert systems. PhD thesis, Department of Computer Science, Boston University, Boston, MA, 1987.
- [61] B Kuipers. Commonsense reasoning about causality: Deriving behavior from structure. In DG Bobrow, Editor, *Qualitative Reasoning about Physical Systems*, pages 169–204. MIT Press, Cambridge, MA, 1985.
- [62] B Kuipers. Qualitative simulation in medical physiology: A progress report. Technical Report TM-280, Laboratory for Computer Science, Massachusetts Institute Technology, Cambridge, MA, 1985.
- [63] B Kuipers. Qualitative simulation of mechanisms. Technical Report TM-274, Laboratory for Computer Science, Massachusetts Institute Technology, Cambridge, MA, 1985.
- [64] B Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289–338, 1986.
- [65] B Kuipers and RS Patil. *Qualitative Simulation and Causal Models*, Volume HA4. Morgan Kaufmann, Los Altos, CA, 1987.
- [66] P Ladkin. Time representation: A taxonomy of interval relations. In *Proceedings AAAI-86*, pages 360–366, Philadelphia, PA, 1986. Morgan Kaufmann.

- [67] P Ladkin. The completeness of a natural system for reasoning with time intervals. In *Proceedings IJCAI-87*, pages 462–467, Milan, Italy, 1987. Morgan Kaufmann.
- [68] P Ladkin. Logical time pieces. *AI Expert*, pages 58–67, 1987.
- [69] P Ladkin. Models of axioms for time intervals. In *Proceedings AAAI-87*, pages 234–239, Seattle, WA, 1987. Morgan Kaufmann.
- [70] CD Lane. The Ozone (O_3) reference manual. Technical Report KSL-86-40, Medical Computer Science Group, Stanford University, Stanford, CA, 1986.
- [71] CP Langlotz, LM Fagan, SW Tu, BI Sikic, and EH Shortliffe. A therapy planning architecture that combines decision theory and artificial intelligence techniques. *Computers and Biomedical Research*, 20:279–303, 1987.
- [72] WJ Long. Reasoning about state from causation and time in the medical domain. In *Proceedings AAAI-83*, pages 251–254, Washington DC, 1983. Morgan Kaufmann.
- [73] WJ Long, S Naimi, MG Criscitiello, and S Kurzrok. Reasoning about therapy from a physiological model. In R Salamon, B Blum, and M Jorgensen, Editors, *MEDINFO 86*, pages 756–760, Washington DC, 1986. North-Holland.
- [74] WJ Long and TA Russ. A control structure for time dependent reasoning. In A Bundy, Editor, *Proceedings IJCAI-83*, pages 230–232, Karlsruhe, West Germany, 1983. Morgan Kaufmann.
- [75] E Mays. A temporal logic for reasoning about changing data bases in the context of natural language question-answering. In L Kershberg, Editor, *Proceedings First International Workshop on Expert Database Systems*, Menlo Park CA, 1984. Benjamin/Cummings.
- [76] E Mays, A Joshi, and B Webber. A modal temporal logic for reasoning about changing databases with applications to natural language questions answering. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1985.
- [77] JM McCarthy. Circumscription—A form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

- [78] JM McCarthy and PJ Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B Meltzer and D Michie, Editors, *Machine Learning 4*. Edinburgh University Press, Edinburgh, England, 1969.
- [79] DV McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.
- [80] CJ McDonald, L Blevins, T Glazener, J Haas, L Lemmon, and J Meeks-Johnson. Regenstrief medical record system. In BI Blum, Editor, *Information Systems for Patient Care*, Chapter 16, pages 232–248. Springer-Verlag, New York, NY, 1984.
- [81] CJ McDonald and WM Tierney. The medical gopher — A microcomputer system to help find, organize and decide about patient data. *Western Journal of Medicine*, 145:823–829, 1986.
- [82] E McKenzie. Bibliography: Temporal databases. *SIGMOD Record*, 15:40–52, 1986.
- [83] KR McKeown. *Text generation: Using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press, Cambridge, England, 1985.
- [84] PL Miller. *A Critiquing Approach to Expert Computer Advice: ATTENDING*. Pitman, Boston, MA, 1984.
- [85] PL Miller and GD Rennels. Prose generation for expert systems: An applied computational linguistics approach. *AI Magazine*, pages 37–44, Fall 1988.
- [86] RA Miller, MA McNeil, SM Challinor, FE Masarie, and JD Myers. The INTERNIST-1/ QUICK MEDICAL REFERENCE project—Status report. *Western Journal of Medicine*, 145:816–822, 1986.
- [87] RA Miller, HE Pople, and JD Myers. INTERNIST-1, An experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine*, 307:468–476, 1982.
- [88] S Mittal. Event-based organization of temporal databases. In *Proceedings Canadian Society of Computational Studies in Intelligence*, pages 164–171, Saskatchewan, Canada, 1982.

- [89] S Mittal and B Chandrasekaran. Conceptual representation of patient data bases. In *Proceedings Hawaii International Conference on System Science*, pages 660–668, Honolulu, HI, 1980.
- [90] S Mittal, B Chandrasekaran, and J Sticklen. Patrec: A knowledge-directed database for a diagnostic expert system. *Computer*, 17:51–58, 1984.
- [91] MA Musen. Generation of model-based knowledge-acquisition tools for clinical-trial advice systems. PhD thesis, Medical Computer Science Group, Stanford University, Stanford, CA, 1988.
- [92] MA Musen, LM Fagan, and EH Shortliffe. Graphical specification of procedural knowledge for an expert system. In *Proceedings IEEE Computer Society Workshop on Visual Languages*, pages 167–178, Dallas, TX, 1986. The Computer Society Press. Reprinted in: *Expert Systems: The User Interface*; J Hendler, Editor; Ablex, Norwood, NJ.
- [93] MA Musen, CP Langlotz, LM Fagan, and EH Shortliffe. Rationale for knowledge base redesign in a medical advice system. In *Proceedings AAMSI Congress 85*, pages 197–201, San Francisco, CA, 1985. American Association for Medical Systems and Informatics.
- [94] A Newell. The knowledge level. *AI Magazine*, 2:1–20, Summer 1981.
- [95] WH Newton-Smith. *The Structure of Time*. Routledge & Kegan Paul, London, England, 1980.
- [96] B Partee. Nominal and temporal anaphora. *Linguistics and Philosophy*, 7:243–286, 1984.
- [97] RS Patil. Causal representation of patient illness for electrolyte and acid–base diagnosis. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1981.
- [98] RS Patil. Review of causal reasoning in medical diagnosis. In *Proceedings Symposium on Computer Applications in Medical Care*, pages 11–15, Washington DC, 1986. IEEE Computer Society.

- [99] SG Pauker, GA Gorry, JP Kassirer, and WB Schwartz. Towards the simulation of clinical cognition taking a present illness by computer. *American Journal of Medicine*, 60:981–996, 1976.
- [100] WH Press, BP Flannery, SA Teukolsky, and WT Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 1986.
- [101] AN Prior. *Time and Modality*. The Clarendon Press, Oxford, England, 1957.
- [102] GD Rennels. A computational model of reasoning from the clinical literature. PhD thesis, Medical Computer Science Group, Stanford University, Stanford, CA, 1986.
- [103] N Rescher and A Urquhart. *Temporal Logic*, Volume 3 of *Library of Exact Philosophy*. Springer-Verlag, Wien, Germany, 1971.
- [104] TA Russ. A system for using time dependent data in patient management. In R Salamon, B Blum, and M Jorgensen, Editors, *MEDINFO 86*, pages 165–169, Washington DC, 1986. North-Holland.
- [105] TA Russ. Temporal control structure reference manual. Technical Report TM-331, Laboratory for Computer Science, Massachusetts Institute Technology, Cambridge, MA, 1987.
- [106] WB Schwartz, RS Patil, and P Szolovits. Artificial intelligence in medicine. Where do we stand? *New England Journal of Medicine*, 316:685–688, 1987.
- [107] LB Sheiner and SL Beal. Bayesian individualization of pharmacokinetics: Simple implementation and comparison with non-Bayesian methods. *Journal of Pharmaceutical Sciences*, 71:1344–1348, 1982.
- [108] Y Shoham. *Reasoning about change*. MIT Press, Cambridge, MA, 1988.
- [109] E Silverberg. Cancer statistics, 1986. *CA—A Cancer Journal for Clinicians*, 36:9–25, 1986.
- [110] H Silverman. A digitalis therapy advisor. PhD thesis, Project MAC, Massachusetts Institute of Technology, Cambridge, MA, 1975.

- [111] BC Smith. Prologue to “Reflection and semantics in a procedural language”. In RJ Brachman and HJ Levesque, Editors, *Readings in Knowledge Representation*, pages 31–39. Morgan Kaufmann, Los Altos, CA, 1985.
- [112] R Snodgrass. Research concerning time in databases: Project summaries. *SIGMOD Record*, 15:19–39, 1986.
- [113] R Snodgrass. The temporal query language TQuel. *ACM Transactions on Database Systems*, 12:247–298, 1987.
- [114] R Snodgrass and I Ahn. A taxonomy of time in databases. In S Navathe, Editor, *Proceedings International Conference on the Management of Data*, pages 236–246, Austin, TX, 1985.
- [115] R Snodgrass and I Ahn. Temporal databases. *IEEE Computer*, 19:35–42, 1986.
- [116] M Stonebraker. *The INGRES papers: Anatomy of a relational database system*. Addison-Wesley, Reading, MA, 1986.
- [117] M Stonebraker, J Anton, and E Hanson. Extending a database system with procedures. *ACM Transactions on Database Systems*, 12:350–376, 1987.
- [118] M Stonebraker, E Wong, P Kreps, and G Held. The design and implementation of INGRES. *ACM Transactions on Database Systems*, 1:189–222, 1976.
- [119] WR Swartout. A digitalis therapy advisor with explanations. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1977.
- [120] P Szolovits, RS Patil, and WB Schwartz. Artificial intelligence in medical diagnosis. *Annals of Internal Medicine*, 108:80–87, 1988.
- [121] EP Tsang. Time structures for AI. In *Proceedings IJCAI-87*, pages 456–461, Milan, Italy, 1987. Morgan Kaufmann.
- [122] DC Tsichritzis and FH Lochovsky. *Data Models*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [123] SW Tu, MG Kahn, MA Musen, JC Ferguson, EH Shortliffe, and LM Fagan. Episodic monitoring of time-oriented data for heuristic skeletal-plan refinement. Technical

Report KSL 87-70, Knowledge Systems Laboratory, Stanford University, Stanford, CA, 1988.

- [124] JD Ullman. *Principles of Database Systems*. Computer Science Press, Rockville, MD, 1982.
- [125] J van Benthem. *The Logic of Time*. D. Reidel, Boston, MA, 1983.
- [126] M Vilain. A system for reasoning about time. In *Proceeding AAAI-82*, pages 197–201, Pittsburgh, PA, 1982. Morgan Kaufmann.
- [127] M Vilain and H Kautz. Constraint propagation algorithms for temporal reasoning. In *Proceedings IJCAI-86*, pages 377–382, Los Angeles, CA, 1986. Morgan Kaufmann.
- [128] B Webber. Special issue: Computational approaches to tense and aspect. *Computational Linguistics*, 14:1–2, 1988.
- [129] S Weyl, J Fries, G Wiederhold, and F Germano. A modular self-describing databank system. *Computers and Biomedical Research*, 8:279–293, 1975.
- [130] QE Whiting-O’Keefe, DW Simborg, WV Epstein, and A Warger. A computerized summary medical record system can provide more information than the standard medical record. *Journal American Medical Association*, 254:1185–1192, 1985.
- [131] LE Widman. Representation method for dynamic causal knowledge using semi-quantitative simulation. In R Salamon, B Blum, and M Jorgensen, Editors, *MED-INFO 86*, pages 180–184, Washington DC, 1986. North-Holland.
- [132] MD Zisman. Use of production systems for modeling asynchronous, concurrent processes. In DA Waterman and F Hayes-Roth, Editors, *Pattern-directed Inference Systems*, pages 53–68. Academic Press, New York, NY, 1978.

APPENDIX A

TQUERY BNF Syntax

TQUERY is the procedural query language used in TNET for specifying date-free, context-dependent temporal queries. The syntax is recursive. Each recursive specification limits the context of the calling query. Recursion is important for limiting the amount of searching through a TNET graph for TNODEs that satisfy a query. The efficiency gains, compared to a linear search through a large TNET, can be substantial. A TQUERY request will return either a list of attribute values that satisfy the query specifications or NIL.

The TQUERY syntax is described in a Backus–Naur Form (BNF) grammar. Two symbols ($::=$ and $|$) are metasymbols of the BNF formalism and are not symbols of TQUERY. The vertical bar denotes different right-hand sides of productions corresponding to the same left-hand side. Bold-faced capitalized strings denote terminal symbols in TQUERY.

<tquery>	::=	(TQUERY <pname> <range> <when> <pcond>)
<pname>	::=	Any valid attribute or parameter name
<range>	::=	<d-range> <i-range> <s-range>
<d-range>	::=	0 1 2 3 4 ...
<i-range>	::=	(<zpos> <zpos>) (<zneg> <zneg>)
<zpos>	::=	0 1 2 3 4 ...
<zneg>	::=	0 -1 -2 -3 -4 ...
<s-range>	::=	ALL FIRST LAST CURRENT OPEN PREVIOUS NIL
<when>	::=	(<i-type> <range> <when> <pname> <pcond>) NIL
<i-type>	::=	<t-label> (<t-label> <st-label>)
<t-label>	::=	Any valid TNODE type label
<st-label>	::=	Any valid TNODE subtype label NIL
<pcond>	::=	Any valid lambda expression containing the replacement character \$

APPENDIX B

ETNET Rule Predicates and Actions

TOPAZ rule predicates examine patient data or model-generated data only within the temporal extent of the ETNODE that contains the rule. The predicate argument WHEN denotes the WHEN syntax, which is presented in detail in Appendix A. In TOPAZ, recursive WHEN specifications were not implemented. In the tables, optional arguments are surrounded by brackets. Nested brackets indicate optional arguments that may appear only if another optional argument is given.

Table B.1 ETNET rule predicates.

Name	Arguments	Interpretation
EXISTS?	TYPE [SUBTYPE] WHEN [PNAME [PVALUE]]	Is there an interval with type label equal to TYPE and subtype label equal to SUBTYPE during WHEN. If PNAME is given, then a satisfying interval must contain some value for PNAME; if PVALUE is also given, then the value of PNAME in any satisfying interval must be equal to PVALUE.
NOTEXISTS?	TYPE [SUBTYPE] WHEN [PNAME [PVALUE]]	Negation of EXISTS? predicate.
BEFORE? AFTER? DURING	INTERVAL WHEN	Is INTERVAL before (after or during) WHEN?
INTERVAL.INRANGE?	TYPE [SUBTYPE] WHEN PNAME LB UB	Is there an interval in WHEN that contains PNAME with a value between LB and UB?
INRANGE?	SOURCE PNAME WHEN LB UB	Is the parameter PNAME's value always between LB and UB during WHEN? SOURCE is MVALUE for model predictions, MPARAM for model parameters, or PVALUE for patient values.
ALMOSTEQ?	SOURCE PNAME WHEN RANGE	Is the parameter PNAME's value during WHEN within RANGE percent of the current value?
SAME.SIGN	SOURCE PNAME WHEN1 WHEN2	Is the sign of parameter PNAME the same during WHEN1 and WHEN2?

Table B.2 ETNET rule actions.

Name	Arguments	Interpretation
NEWINTERVAL	TYPE [SUBTYPE] START [STOP]	Create a new ETNODE interval. If no STOP is specified, set STOP value to <i>future</i> .
ASSERT.MIN.VALUE ASSERT.MAX.VALUE ASSERT.MEAN.VALUE ASSERT.SIZE	INTERVAL SOURCE PNAME	Assert into INTERVAL the minimum (maximum, mean, or size=max – min) value of PNAME found within the temporal extent of INTERVAL.
ASSERT.SYMBOLIC.VALUE	INTERVAL SOURCE PNAME PVALUE	Assert the PVALUE for the parameter PNAME into INTERVAL.
ASSERT.DURATION	INTERVAL	Assert INTERVAL's duration into INTERVAL.
ASSERT.REMARKABLE	INTERVAL	Assert the REMARKABLE flag into INTERVAL. This flag denotes intervals whose presence has domain-specific interest.

APPENDIX C

TOPAZ-Generated Patient Summaries

In the following pages, I present six patient records that were summarized by TOPAZ. For each patient, I include

1. The portion of the ONCOCIN flowsheet that contains the observed WBC counts and chemotherapy doses
2. A graphical display of the ONCOCIN TNET (Chapter 4) representation of the therapeutic events
3. A graphical display of the TOPAZ ETNET (Chapter 7) representation of abstracted clinical findings
4. The full summary text generated by the TOPAZ text generation system (Chapter 8)

Figures 1.1 (page 3) and 1.3 (page 14) were generated from patient 6. Figures 5.6 (page 135) and 5.7 (page 136) were generated from patient 1. Patient 2 had the most typical clinical course, patient 3 had the most atypical clinical course, and patient 4 had the most problems with potential model-misfitting.

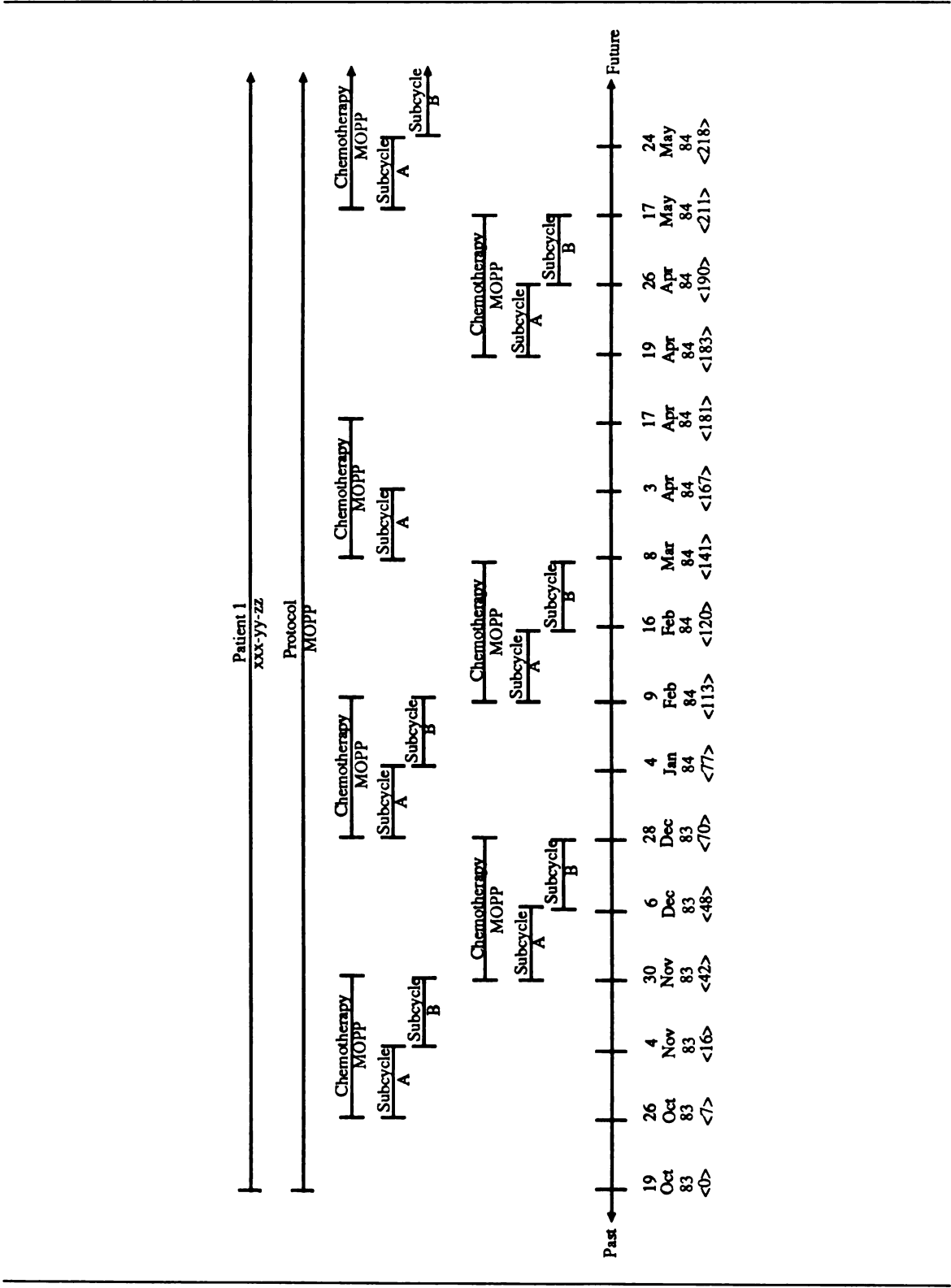
Upper Nodes	2		2		1	1		1	1	
Lower Nodes	1		1		1	1		1	1	
Spleen										
Splenectomy										
Liver Bx										
Marrow Bx	1									
LAG	2					2		2		
IVP										
CXR	6					2		2		
Bone Scan										
CT Scan										
OVERALL	6			3	3	2		2	2	
Karnofsky	100			100	100	100		100	100	
Weight	90.7		89.5	88.2	87.4	89.1		87.3	88.5	84

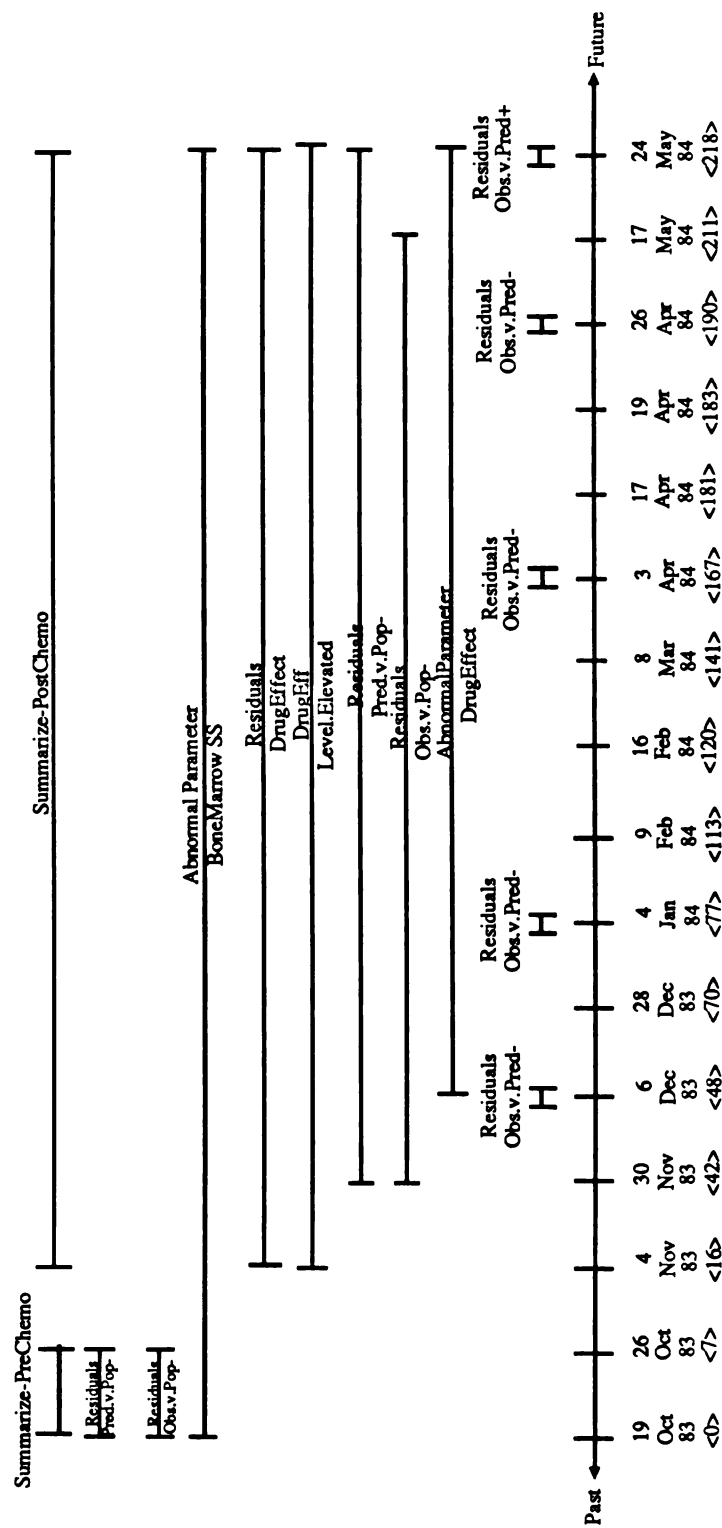
WBC X 1000	7.7	7.9	6.9	4.8	3.1	4.4	3.3	4.7	5.3	5.0
PCV	40.9	40.4	37.1	36.3	32.2	32.6	28.2	27.7	24.4	29.9
Hemoglobin	13.1	12.8	12.2	11.8	10.6	10.5	9.4	8.8	9.4	9.2
Platelets	574	329	368	282	305	120	112	193	150	106

BSA (m ²)		2.11								
Chemo Name		MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP
Cycle #		1	1	2	2	3	3	4	4	5
Subcycle		A	B	A	B	A	B	A	B	A
Visit Type	LAB	TX	TX	TX	TX	TX	TX	TX	TX	TX
Mustard		12	12	12	12	10	8	8	8	8
Vincristine		2.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0	1.0
Procarbazine		200	200	200	100	75	75	75	75	75
Prednisone										

Port										
RTx Start Date										
RTx End Date										
Cum. # Fraction										
Cum. Max. Dose										
Number Rads										
RT Toxicity										

Day	19	26	4	30	6	28	4	9	16	8
Month	Oct	Oct	Nov	Nov	Dec	Dec	Jan	Feb	Feb	Mar
Year	83	83	83	83	83	83	84	84	84	84





Patient Identifying Data:

This is a 25 y/o male who presented with clinical stage III-S-A, pathological stage III-S-A Hodgkins' lymphoma. The patient's medical record begins on 19 October 1983 and ends on 24 May 1984. He was assigned to MOPP chemotherapy and received 7 treatment cycles.

Analysis of Clinical Course:

CAUTION: This analysis is based on models of the effects of MOPP chemotherapy on bone marrow and WBC kinetics to derive inferences about unobserved physiological processes. Model mis-specification and/or measurement inaccuracies can result in serious errors in the conclusions derived from these models. In the analysis that follows, conclusions that are based on periods in the record when the model fit appears to be poor are noted. Please take these limitations into account while reading this analysis.

Based on our analysis of the entire clinical course, the best fit of our model to the data suggests the following key patient-specific physiological parameter values:

1. A steady-state bone marrow mass that is approximately 0.6 times the population value. This is considered a moderate reduction in steady-state bone marrow mass.
2. A duration of drug effect that is approximately 2.2 times the population value. This is considered a moderate increase in the duration of drug effect.
3. A sensitivity to drug effect that is approximately 1.4 times the population value. This is considered a mild increase in the sensitivity to drug effect.

Based on differences between population-based predictions (the expected clinical course) and the patient-specific model predictions (the modeled clinical course), the patient had 2 remarkable clinical events. For each event, we provide a description of the event, a probable cause for the event, and an analysis of the ability of the patient-specific model to predict the actual observations during this event.

1. A series of 2 visits prior to receiving chemotherapy, starting on 19 October 1983 and ending on 26 October 1983, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 26 October 1983 (63% of the expected value).

The patient's estimated steady-state marrow mass was lower than expected causing a lower than expected pre-treatment WBC count. The model cannot explain the underlying etiology of a lower than expected steady state marrow mass.

There were no periods of poor model fit during this remarkable clinical event.

2. A series of 13 visits after starting chemotherapy, starting on 30 November 1983 and ending on 24 May 1984, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 28 December 1983 (64% of the expected value).

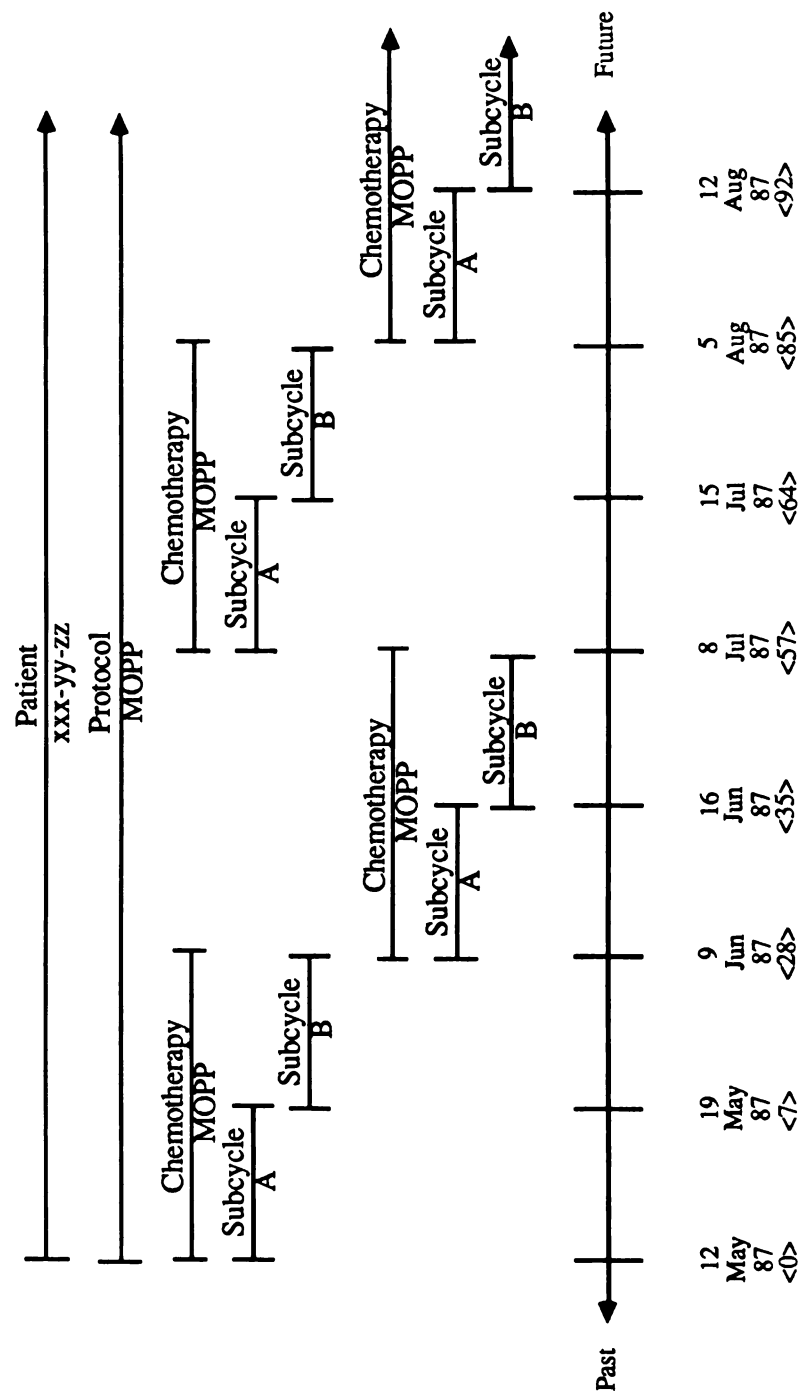
After adjusting for the patient's steady-state bone marrow mass, the two drug-related parameters, duration of drug effect and drug sensitivity, jointly caused an increased drug effect which led to increased myelosuppression and lower than expected WBC

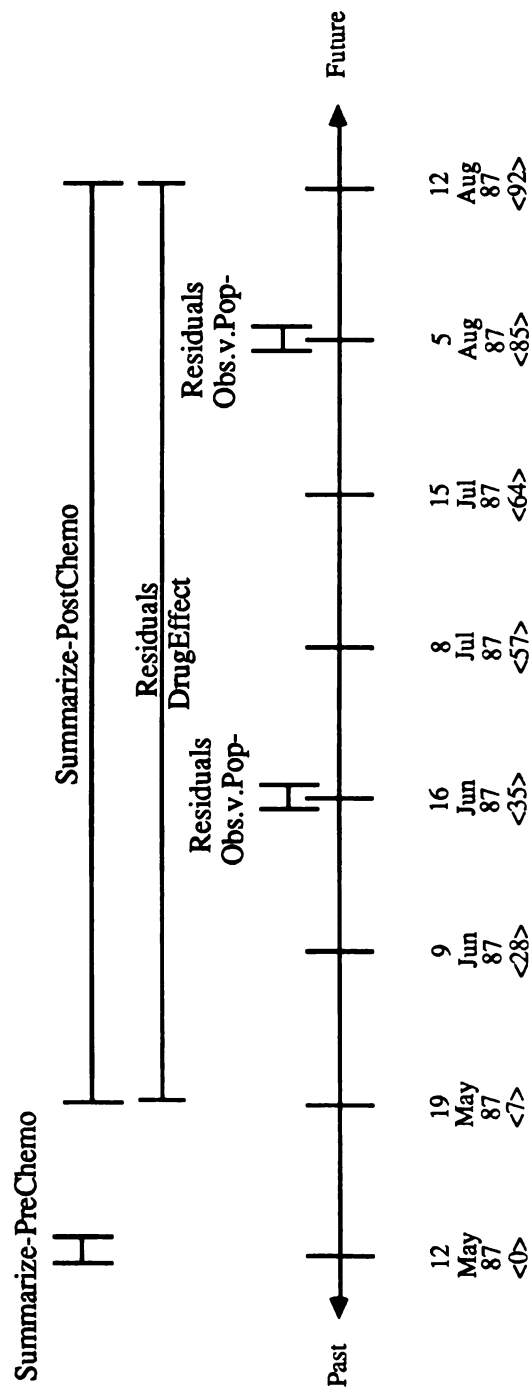
counts. The increased drug effect had a peak myelosuppressive effect 13.1 times greater than expected on 19 April 1984. The accumulation of drug due to the prolonged duration of drug effect was responsible for a 9.4 fold increase in drug effect. The increased sensitivity to drug was responsible for a 1.4 fold increase in drug effect.

Out of the remaining 13 visits during the period, there were 5 episodes of nonsystematic model versus observation misfitting (6 December 1983, 4 January 1984, 3 April 1984, 26 April 1984, and 24 May 1984). Because these model misfits were not sustained for longer than one observation, we feel they do not alter the parameter estimates derived over this period.

There were no episodes of model misfitting that occurred during the unremarkable periods in the patient's clinical course.

[illegible]





Patient Identifying Data:

This is a 40 y/o male who presented with clinical stage II-E-A, pathological stage II-E-A Hodgkins' lymphoma. The patient's medical record begins on 12 May 1987 and ends on 12 August 1987. He was assigned to MOPP chemotherapy and received 4 treatment cycles.

Analysis of Clinical Course:

CAUTION: This analysis is based on models of the effects of MOPP chemotherapy on bone marrow and WBC kinetics to derive inferences about unobserved physiological processes. Model mis-specification and/or measurement inaccuracies can result in serious errors in the conclusions derived from these models. In the analysis that follows, conclusions that are based on periods in the record when the model fit appears to be poor are noted. Please take these limitations into account while reading this analysis.

Based on our analysis of the entire clinical course, the best fit of our model to the data suggests the following key patient-specific physiological parameter values:

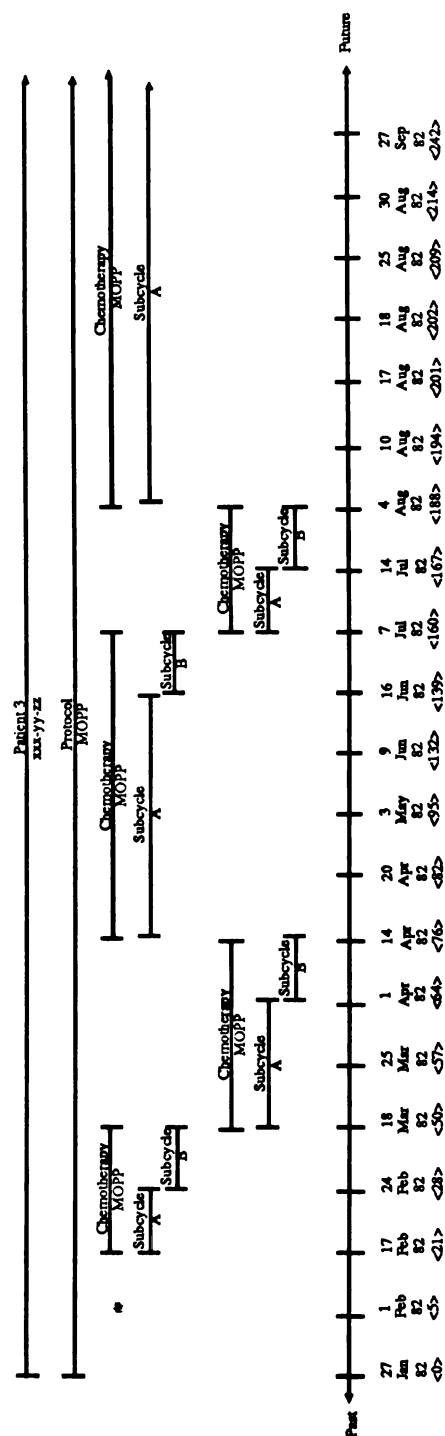
1. A steady-state bone marrow mass that is approximately 0.9 times the population values. This is considered a mild reduction in the steady-state bone marrow mass.
2. A duration of drug effect that is approximately 1.2 time the population value. This is considered a mild increase in the duration of drug effect.
3. A normal sensitivity to drug effect.

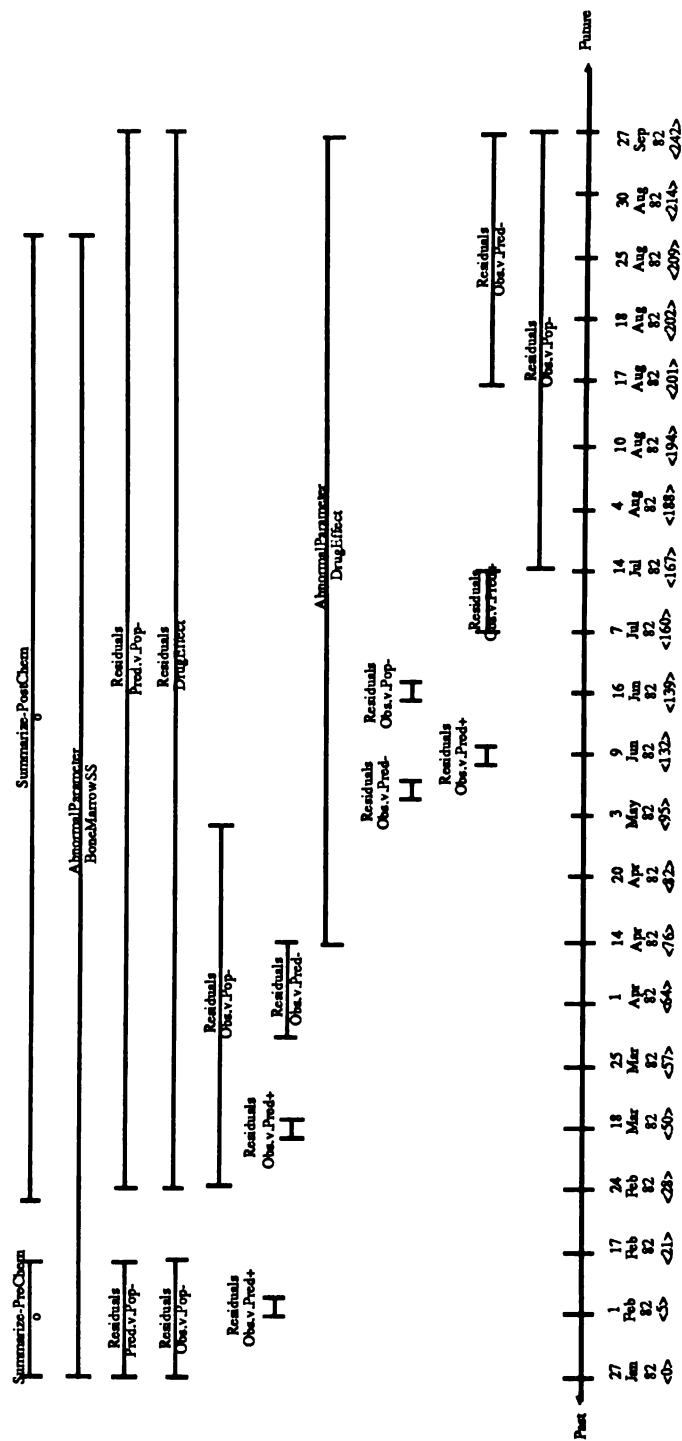
Based on differences between population-based predictions (the expected clinical course) and the patient-specific model predictions (the modeled clinical course), the patient had no remarkable clinical events.

There were no episodes of model misfitting that occurred during the unremarkable periods in the patient's clinical course.

[illegible]

[illegible]





Patient Identifying Data:

This is a 28 y/o male who presented with unknown clinical stage, pathological stage III-E-A Hodgkins' lymphoma. The patient's medical record begins on 27 January 1982 and ends on 27 September 1982. He was assigned to MOPP chemotherapy and received 5 treatment cycles.

Analysis of Clinical Course:

CAUTION: This analysis is based on models of the effects of MOPP chemotherapy on bone marrow and WBC kinetics to derive inferences about unobserved physiological processes. Model mis-specification and/or measurement inaccuracies can result in serious errors in the conclusions derived from these models. In the analysis that follows, conclusions that are based on periods in the record when the model fit appears to be poor are noted. Please take these limitations into account while reading this analysis.

Based on our analysis of the entire clinical course, the best fit of our model to the data suggests the following key patient-specific physiological parameter values:

1. A steady-state bone-marrow mass that is approximately 0.4 times the population value. This is considered a marked reduction in the steady-state bone-marrow mass.
2. A duration of drug effect that is approximately 2.5 times the population value. This is considered a moderate increase in the duration of drug effect.
3. A sensitivity to drug effect that is approximately 1.3 times the population value. This is considered a mild increase in the sensitivity to drug effect.

Based on differences between population-based predictions (the expected clinical course) and the patient-specific model predictions (the modeled clinical course), the patient had 2 remarkable clinical events. For each event, we provide a description of the event, a probable cause for the event, and an analysis of the ability of the patient-specific model to predict the actual observations during this event.

1. A series of 3 visits prior to receiving chemotherapy, starting on 27 January 1982 and ending on 17 February 1982, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 17 February 1982 (43 per cent of the expected value).

The patient's estimated steady-state marrow mass was lower than expected causing a lower than expected pre-treatment WBC count. The model cannot explain the underlying etiology of a lower than expected pre-treatment steady-state bone-marrow mass.

Out of the remaining 3 visits during this period, there was 1 episode of nonsystematic model versus observation misfitting (1 Feb 82). Because this model misfit was not sustained for longer than one observation, we feel it does not alter the parameter estimates derived over this period.

2. A series of 18 visits after starting chemotherapy, starting on 24 February 1982 and ending on 27 September 1982, with predicted WBC counts that were systematically

lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 4 August 1982 (60% of the expected value).

After adjusting for the patient's steady-state bone-marrow mass, the two drug-related parameters, *duration of drug effect* and *drug sensitivity*, jointly caused an increased drug effect which lead to increased myelosuppression and lower than expected WBC counts. The increased drug effect had a peak myelosuppressive effect 2.9 times greater than expected on 1 April 1982. The accumulation of drug due to the prolonged duration of drug effect was responsible for a 2.2 fold increase in drug effect. The increased sensitivity to drug was responsible for a 1.3 fold increase in drug effect.

From 1 April 1982 until 14 April 1982 (2 out of 18 visits), there was a period during which the model predictions were systematically closer to the expected values than were the actual observations. This difference means that the patient's true parameter values may have been further from normal than we have estimated for this period.

From 7 July 1982 until 14 July 1982 (2 out of 18 visits), there was a period during which the model predictions were systematically closer to the expected values than were the actual observations, This difference means that the patient's true parameter values may have been further from normal than we have estimated for this period.

From 17 August 1982 until 27 September 1982 (5 out of 18 visits), there was a period during which the model predictions were systematically close to the expected values than were the actual observations, This difference means that the patient's true parameter values may have been further from normal than we have estimated for this period.

Out of the remaining 9 visits during this period, there were 3 episodes of nonsystematic model versus observation misfitting (18 Mar 82, 3 May 82, and 9 Jun 82). Because these model misfits were not sustained for longer than one observation, we feel they do not alter the parameter estimates derived over this period.

There were no episodes of model misfitting that occurred during the unremarkable periods in the patient's clinical course.

•

[illegible]

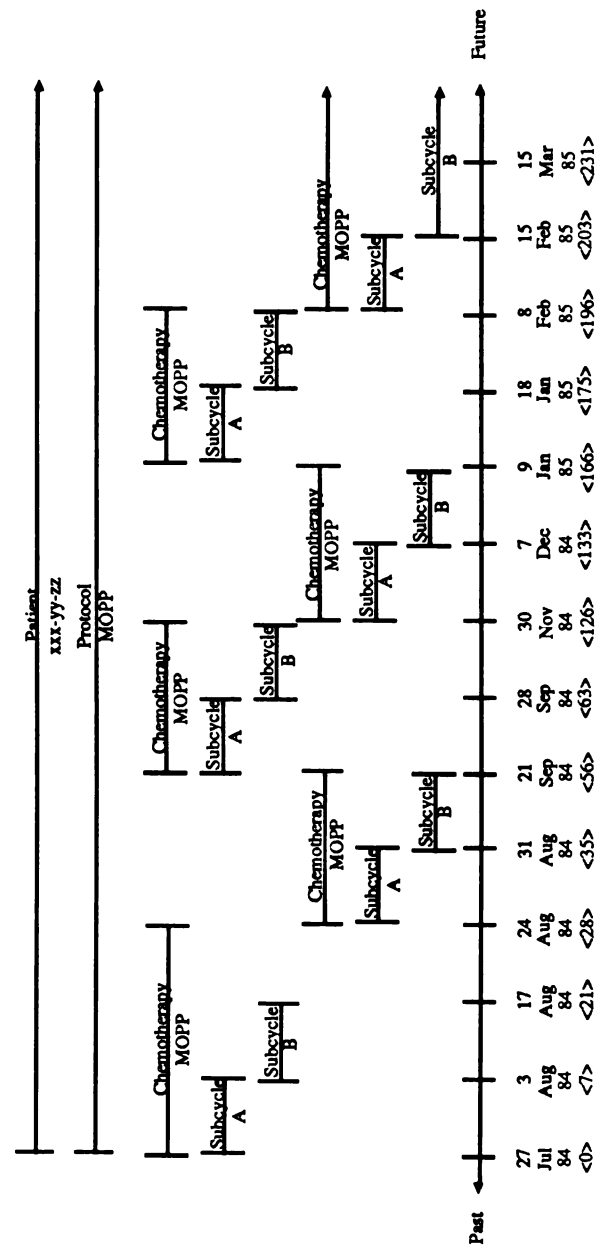
Upper Nodes	1	1	1	1						
Lower Nodes	1	1	1	1						
Spleen	1	1	1	1						
Splenectomy										
Liver Bx										
Marrow Bx										
LAG										
IVP										
CXR	2									
Bone Scan										
CT Scan										
OVERALL	2	2	2	2						
Karnofsky	80	60		70						
Weight		73.9								

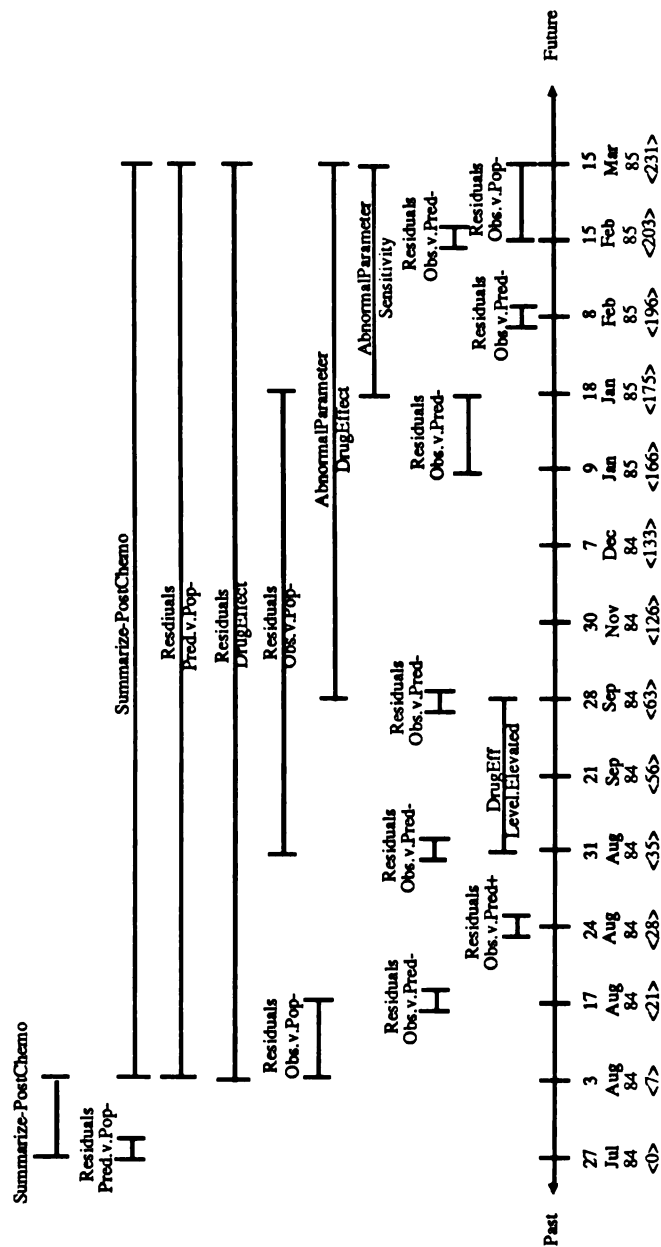
WBC X 1000	2.4	7.9	2.5	6.7						
PCV	38.3	36.4	31.6	33						
Hemoglobin	13.3	12.6	10.7	10.8						
Platelets	102	128	74	105						

BSA (m ²)										
Chemo Name	MOPP	MOPP	MOPP							
Cycle #	5	6	6							
Subcycle	B	A	B							
Visit Type	TX	TX	TX	LAB						
Mustard	3	11	3							
Vincristine	0	0	0							
Procarbazine	200	0	50							
Prednisone										

Port										
RTx Start Date										
RTx End Date										
Cum. # Fraction										
Cum. Max. Dose										
Number Rads										
RT Toxicity										

Day	18	8	15	15						
Month	Jan	Feb	Feb	Mar						
Year	85	85	85	85						





Patient Identifying Data:

This is a 43 y/o male who presented with clinical stage II-A, pathological stage II-A Hodgkins' lymphoma. The patient's medical record begins on 27 July 1984 and ends on 15 March 1985. He was assigned to MOPP chemotherapy and received 6 treatment cycles.

Analysis of Clinical Course:

CAUTION: This analysis is based on models of the effects of MOPP chemotherapy on bone marrow and WBC kinetics to derive inferences about unobserved physiological processes. Model mis-specification and/or measurement inaccuracies can result in serious errors in the conclusions derived from these models. In the analysis that follows, conclusions that are based on periods in the record when the model fit appears to be poor are noted. Please take these limitations into account while reading this analysis.

Based on our analysis of the entire clinical course, the best fit of our model to the data suggests the following key patient-specific physiological parameter values:

1. A steady-state bone marrow mass that is approximately 0.8 times the population values. This is considered a mild reduction in the steady-state bone marrow mass.
2. A duration of drug effect that is approximately 2.1 time the population value. This is considered a moderate increase in the duration of drug effect.
3. A sensitivity to drug effect that is approximately 1.8 times the population value. This is considered a mild increase in the sensitivity to drug effect.

Based on differences between population-based predictions (the expected clinical course) and the patient-specific model predictions (the modeled clinical course), the patient had 2 remarkable clinical events. For each event, we provide a description of the event, a probable cause for the event, and an analysis of the ability of the patient-specific model to predict the actual observations during this event.

1. A single visit prior to receiving chemotherapy, on 27 July 1984, with the predicted WBC count lower than expected on that date (83 percent of the expected value).

The patient's estimated steady-state bone-marrow mass was lower than expected causing a lower than expected pretreatment WBC count. The model cannot explain the underlying etiology of a lower than expected steady-state marrow mass.

There were no periods of poor model fitting during this remarkable clinical event.

2. A series of 13 visits after starting chemotherapy, starting on 3 August 1984 and ending on 15 March 1985, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 28 September 1984 (50 percent of the expected value).

After adjusting for the patient's steady-state bone-marrow mass, the two drug-related parameters, duration of drug effect and drug sensitivity, jointly caused an increased drug effect which led to increased myelosuppression and lower than expected WBC counts. The increased drug effect had a peak myelosuppressive effect 3.7 times greater than expected on 28 September 1984. The accumulation of drug due to the prolonged

duration of drug effect was responsible for a 2.1 fold increased in drug effect. The increased sensitivity to drug was responsible for a 1.8 fold increase in drug effect.

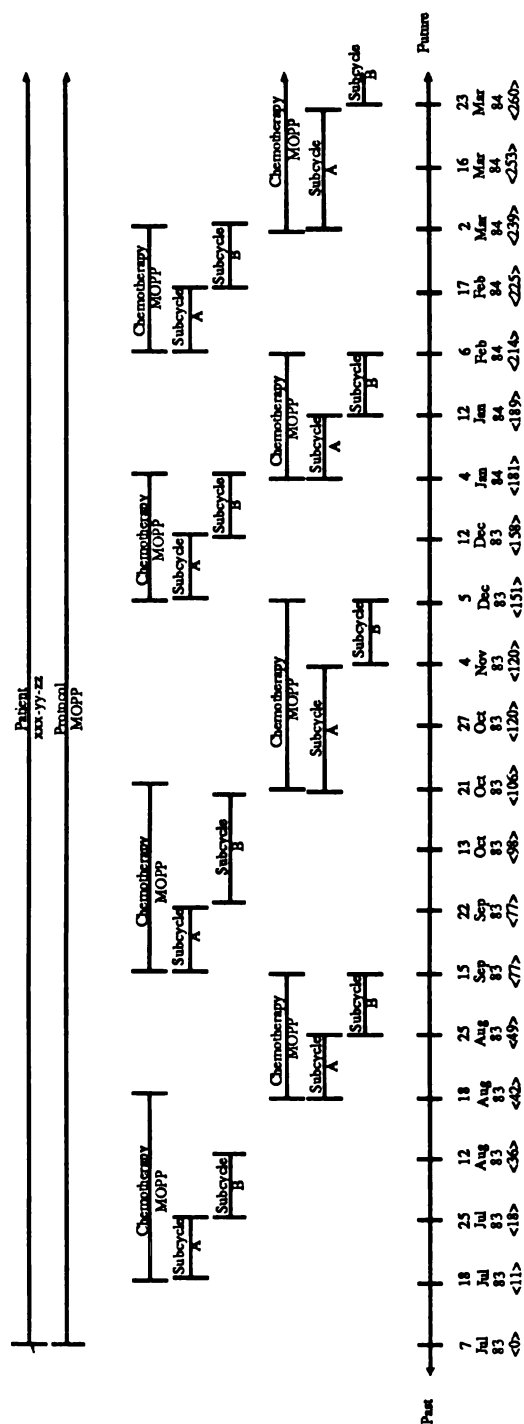
From 9 January 1985 until 18 Jan 1985 (2 out of 13 visits), there was a period during which the model predictions were systematically closer to the expected values than were the actual observations. This difference means that the patient's true parameter values may have been further from normal than we have estimated for this period.

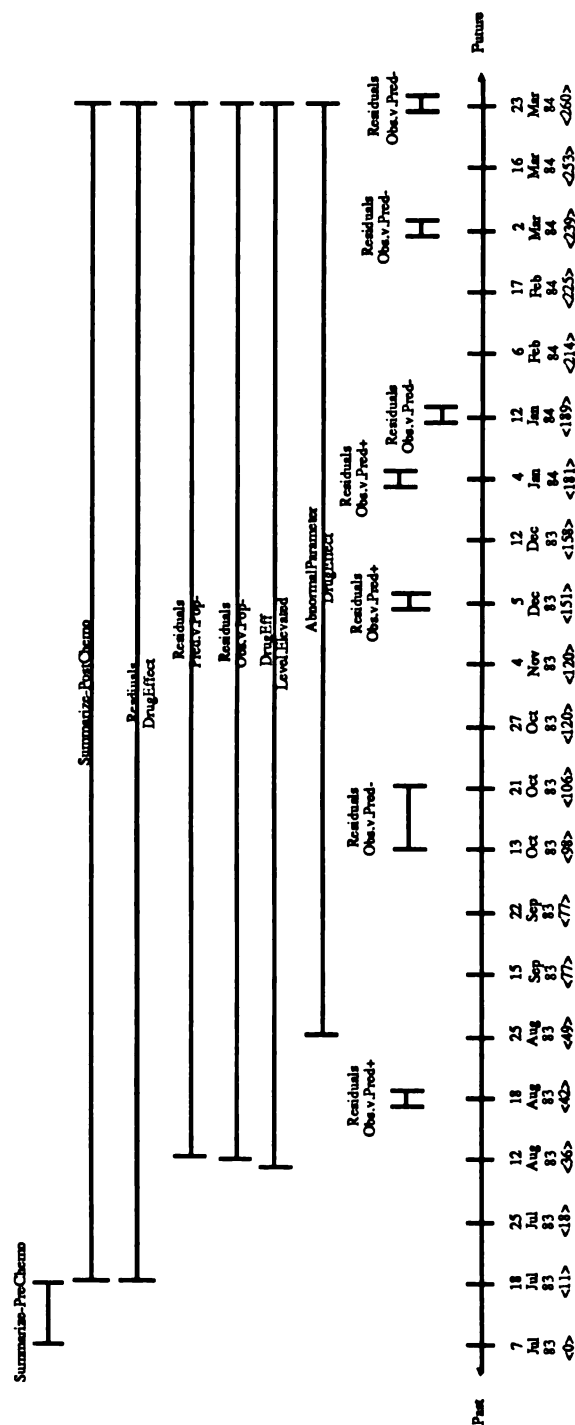
Out of the remaining 11 visits during this period, there were 6 episodes of nonsystematic model versus observation misfitting (17-Aug-84, 24-Aug-84, 31-Aug-84, 28-Sep-84, 8-Feb-85, and 15-Feb-85). Because these model misfits were not sustained for longer than one observation, we feel they do not alter the parameter estimates derived over this period.

There were no episodes of model misfitting that occurred during the unremarkable periods in the patient's clinical course.

[illegible]

Upper Nodes	1		1		1					
Lower Nodes	1		1		1					
Spleen										
Splenectomy										
Liver Bx										
Marrow Bx										
LAG	3		1	2			1			
IVP										
CXR	1						1			
Bone Scan										
CT Scan										
OVERALL	3			2	2	2				
Karnofsky	90			100	100	100				
Weight	82.5	79.3	79.3	77.3	78.2			76.7		
WBC X 1000	3.5	3.5	6.3	3.4	4.6	1.6	4.7	5.1	3.3	4.3
PCV	32.8	32.3	33.4	31.2	34.5	31.0	39.8	34.8	35.7	39.4
Hemoglobin	10.4	10.8	10.8	10.4	11.6	10.3	11.9	11.6	12.1	12.9
Platelets	405	378	273	238	316	281	317	327	240	204
BSA (m ²)										
Chemo Name	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP	MOPP
Cycle #	4	4	5	5	6	6	6	6	7	7
Subcycle	A	B	A	B	A	B	A	B	A	A
Visit Type	TX	TX	TX	TX	TX	ABORT	TX	TX	DELAY	TX
Mustard	8	8	10	7	8		6	6		6
Vincristine	1	0	0	0	0		0	0		0
Procarbazine	100	100	100	50	100		75	75		75
Prednisone										
Port										
RTx Start Date										
RTx End Date										
Cum. # Fraction										
Cum. Max. Dose										
Number Rads										
RT Toxicity										
Day	27	4	5	12	4	12	6	17	2	16
Month	Oct	Nov	Dec	Dec	Jan	Jan	Feb	Feb	Mar	Mar
Year	83	83	83	83	84	84	84	84	84	84





Patient Identifying Data:

This is a 42 y/o male who presented with clinical stage II-A, pathological stage II-S-A Hodgkins' lymphoma. The patient's medical record begins on 7 July 1983 and ends on 23 March 1984. He was assigned to MOPP chemotherapy and received 8 treatment cycles.

Analysis of Clinical Course:

CAUTION: This analysis is based on models of the effects of MOPP chemotherapy on bone marrow and WBC kinetics to derive inferences about unobserved physiological processes. Model mis-specification and/or measurement inaccuracies can result in serious errors in the conclusions derived from these models. In the analysis that follows, conclusions that are based on periods in the record when the model fit appears to be poor are noted. Please take these limitations into account while reading this analysis.

Based on our analysis of the entire clinical course, the best fit of our model to the data suggests the following key patient-specific physiological parameter values:

1. A normal bone-marrow steady-state mass.
2. A duration of drug effect that is approximately 6.4 times the population value. This is considered a marked increase in the duration of drug effect.
3. A sensitivity to drug effect that is approximately 1.6 times the population value. This is considered a mild increase in the sensitivity to drug effect.

Based on differences between population-based predictions (the expected clinical course) and the patient-specific model predictions (the modeled clinical course), the patient had 1 remarkable clinical event. For this event, we provide a description of the event, a probable cause for the event, and an analysis of the ability of the patient-specific model to predict the actual observations during this event.

1. A series of 18 visits after starting chemotherapy, starting on 12 August 1983 and ending on 23 March 1984, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 5 December 1983 (37 percent of the expected value.)

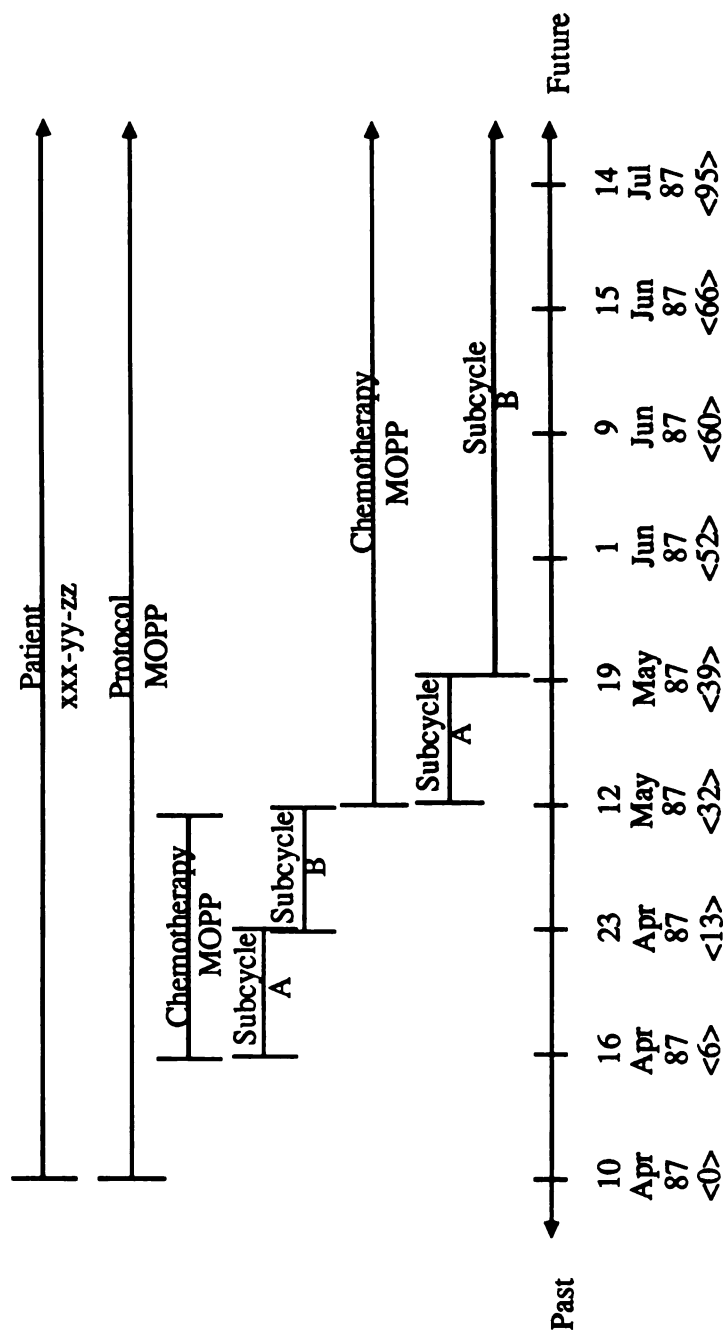
After adjusting for the patient's steady-state bone-marrow mass, the two drug-related parameters, duration of drug effect and drug sensitivity, jointly caused an increased drug effect which lead to increased myelosuppression and lower than expected WBC counts. The increased drug effect had a peak myelosuppressive effect 10.5 times greater than expected on 12 January 1984. The accumulation of drug due to the prolonged duration of drug effect was responsible for a 6.6 fold increase in drug effect. The increased sensitivity to drug was responsible for a 1.6 fold increase in drug effect.

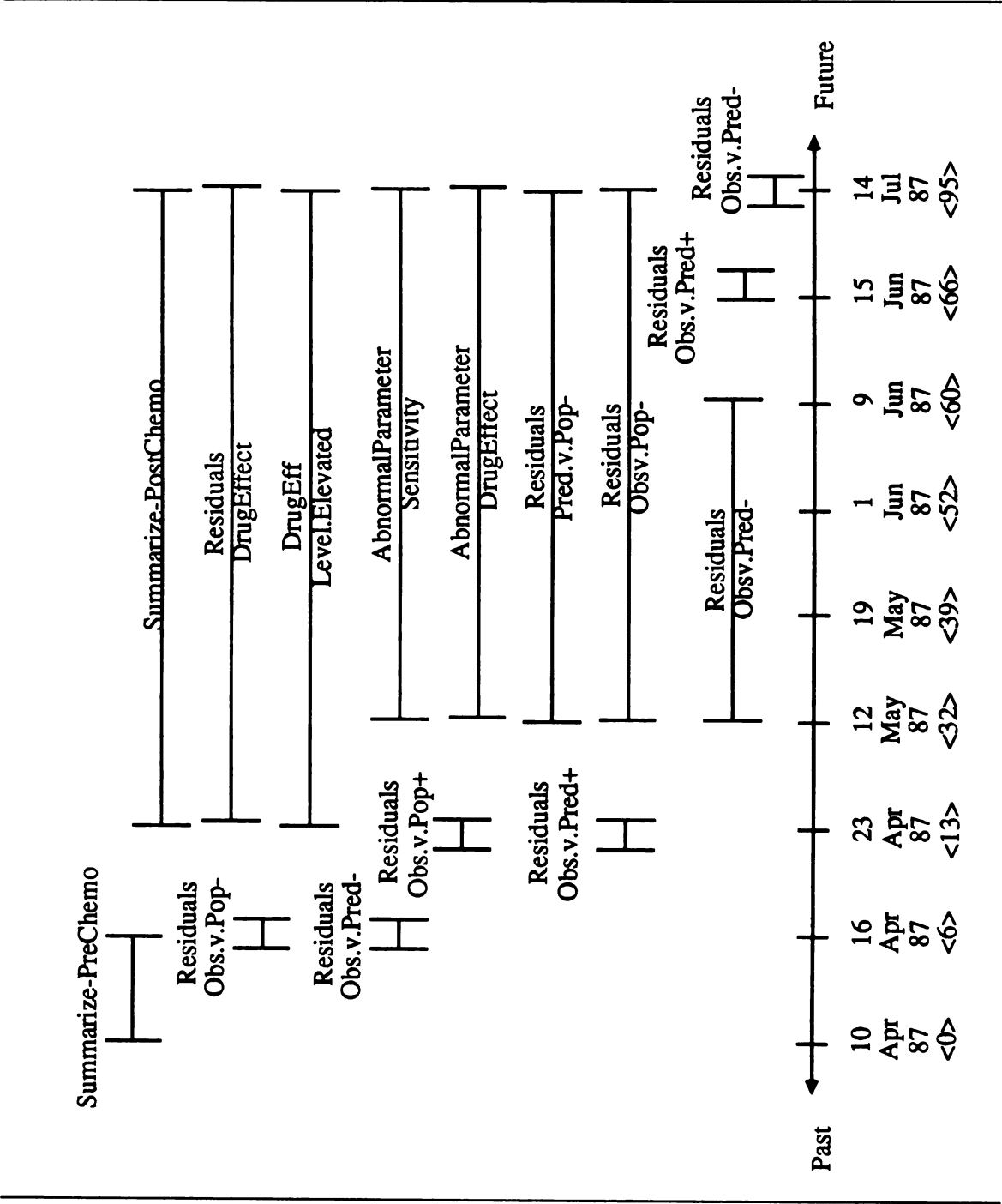
From 13 October 1983 until 21 October 1983 (2 out of 18 visits), there was a period during which the model predictions were systematically closer to the expected values than were the actual observations. This difference means that the patient's true parameter values may have been further from normal than we have estimated for this period.

Out of the remaining 16 visits during this period, there were 6 episodes of nonsystematic model versus observation misfitting (18-Aug-83, 5-Dec-83, 4-Jan-84, 12-Jan-84, 2-Mar-84, and 23-Mar-84). Because these model misfits were not sustained for longer than one observation, we feel they do not alter the parameter estimates derived over this period.

There were no episodes of model misfitting that occurred during the unremarkable periods in the patient's clinical course.

[illegible]





Patient Identifying Data:

This is a 21 y/o female who presented with clinical stage III-E-E-A, pathological stage I-A Hodgkins' lymphoma. The patient's medical record begins on 10-Apr-87 and ends on 14-Jul-87. She was assigned to MOPP chemotherapy and received 2 treatment cycles.

Analysis of Clinical Course:

CAUTION: This analysis is based on models of the effects of MOPP chemotherapy on bone marrow and WBC kinetics to derive inferences about unobserved physiological processes. Model mis-specification and/or measurement inaccuracies can result in serious errors in the conclusions derived from these models. In the analysis that follows, conclusions that are based on periods in the record when the model fit appears to be poor are noted. Please take these limitations into account while reading this analysis.

Based on our analysis of the entire clinical course, the best fit of our model to the data suggests the following key patient-specific physiological parameter values:

1. A steady-state bone marrow mass that is approximately 0.9 times the population value. This is considered a mild reduction in steady-state bone marrow mass.
2. A duration of drug effect that is approximately 4.2 times the population value. This is considered a moderate increase in the duration of drug effect.
3. A sensitivity to drug effect that is approximately 2.4 times the population value. This is considered a moderate increase in the sensitivity to drug effect.

Based on differences between population-based predictions (the expected clinical course) and the patient-specific model predictions (the modeled clinical course), the patient had one remarkable clinical event. For each event, we provide a description of the event, a probable cause for the event, and an analysis of the ability of the patient-specific model to predict the actual observations during this event.

1. A series of 6 visits after starting chemotherapy, starting on 12 May 1987 and ending on 14 July 1987, with predicted WBC counts that were systematically lower than expected for this period. The predicted WBC count with the largest deviation from expected occurred on 15 June 1987 (35% of the expected value).

After adjusting for the patient's steady-state bone marrow mass, the two drug-related parameters, duration of drug effect and drug sensitivity, jointly caused an increased drug effect which led to increased myelosuppression and lower than expected WBC counts. The increased drug effect had a peak myelosuppressive effect 8.7 times greater than expected on 1 June 1987. The accumulation of drug due to the prolonged duration of drug effect was responsible for a 3.6 fold increase in drug effect. The increased sensitivity to drug was responsible for a 2.4 fold increase in drug effect.

From 12 May 1987 until 9 June 1987 (4 out of 6 visits), there was a period during which the model predictions were systematically closer to the expected values than were the actual observations. This difference means that the patient's true parameter values may have been further from normal than we have estimated for this period.

Out of the remaining 2 visits during the period, there were 2 episodes of nonsystematic model versus observation misfitting (15 June 1987 and 14 July 1987). Because these model misfits were not sustained for longer than one observation, we feel they do not alter the parameter estimates derived over this period.

There were 2 episodes of model misfitting that did not occur during any remarkable clinical event:

1. On 16 April 1987, there was an isolated unusual observation that was poorly fitted by the model. This observation was not part of a period of sustained model misfitting.
2. On 23 April 1987, there was an isolated unusual observation that was poorly fitted by the model. This observation was not part of a period of sustained model misfitting.

FOR REFERENCE

NOT TO BE TAKEN FROM THE ROOM



CAT. NO. 23 012

PRINTED
IN
U.S.A.

