PROCESS-ORIENTED ANALYSIS AND DISPLAY OF CLINICAL LABORATORY DATA

by

Andrew Ronald Post

BA, Biochemistry, University of Pennsylvania, 1994

MD, University of Pennsylvania, 1999

MS, Information Science, University of Pittsburgh, 2001

Submitted to the Graduate Faculty of

Medicine in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2006

UNIVERSITY OF PITTSBURGH

SCHOOL OF MEDICINE

This dissertation was presented

by

Andrew Ronald Post

It was defended on

October 27, 2006

and approved by

Gregory Cooper, MD, PhD, Associate Professor

Vanathi Gopalakrishnan, PhD, Assistant Professor

Milos Hauskrecht, PhD, Assistant Professor

Valerie Monaco, PhD, MHCI, Assistant Professor
Dissertation Co-director

James H. Harrison, Jr., MD, PhD, Associate Professor
Dissertation Director

PROCESS-ORIENTED ANALYSIS AND DISPLAY OF CLINICAL LABORATORY DATA

Andrew Ronald Post, MD, PhD

University of Pittsburgh, 2006

Background: Disease and patient care processes often create characteristic mathematical and temporal patterns in time-stamped clinical events and observations, but existing medical record systems have a limited ability to recognize or visualize these patterns.

System Design: This dissertation introduces the process-oriented approach to clinical data analysis and visualization. This approach aims to support specifying, detecting, and visualizing mathematical and temporal patterns in time-stamped patient data for a broad range of clinical tasks. It has two components: a pattern specification and detection strategy called PROTEMPA (Process-oriented Temporal Analysis); and a pattern visualization strategy called TPOD (Temporal Process-oriented Display).

Evaluation: A study in the clinical research domain evaluated PROTEMPA's ability to identify and categorize patients based on diagnosis, disease severity, and disease progression by scanning for patterns in clinical laboratory results. A cognitive study in the patient care domain evaluated PROTEMPA and TPOD's ability to help physicians review cases and make decisions using case presentation software that displays laboratory results in either a TPOD-based display or a standard laboratory display.

Results: PROTEMPA successfully identified laboratory data patterns in both domains. TPOD successfully visualized these patterns in the patient care domain. In the patient care study,

subjects obtained more clinical concepts from the TPOD-based display, but TPOD had no effect on decision-making speed or quality. Subjects were split on which laboratory display they preferred, but expressed a desire to gain more familiarity with the TPOD-based display. Subjects reviewed data in the standard laboratory display for a variety of purposes, and interacted with the display in a complex fashion.

Conclusions: The process-oriented approach successfully recognized and visualized data patterns for two distinct clinical tasks. In clinical research, this approach may provide significant advantages over existing methods of data retrieval. In patient care, comparative evaluation of novel data displays in context provides insights into physicians' preferences, the process of clinical decision-making by physicians, and display usability. TPOD's influence on concept acquisition is promising, but further research is needed regarding physicians' use of laboratory data for results review in order to determine how a process-oriented display might be deployed most beneficially.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

To my wife, Anh, I can't thank you enough for your love and support, for hanging in there through the long hours I've spent hunched over a computer, and for encouraging me to embark on a research career instead of just doing a residency after medical school like everyone else. I'd like to thank my three-month-old son Ivan, whose social smiles have helped me get through the home stretch of finishing my dissertation. I'd also like to thank John Nguyen, who introduced me to and sparked my interest in biomedical informatics.

I'd like to thank the Department of Biomedical Informatics for providing such a fantastic environment for learning about this field. Special thanks to my academic and research advisor, Dr. James Harrison, who supported my ideas from the beginning; the other members of my committee, Dr. Gregory Cooper, Dr. Vanathi Gopalakrishnan, Dr. Milos Hauskrecht, and Dr. Valerie Monaco for their help and feedback; Dr. Charles Friedman, for convincing me to do a PhD and for continuing to be a mentor even after going on sabbatical; and Dr. Cynthia Gadd, for making the process of going through the new PhD program as smooth as possible.

Finally, I'd like to thank the National Library of Medicine for so generously supporting my training (training grant T15 LM007059), and for continuing to support the research in this dissertation through grant R01 LM008192.

1.0 INTRODUCTION

Health care institutions store large volumes of clinical data that are useful for understanding processes associated with disease, therapeutic response, and patient care. These processes may be reflected as time sequences of laboratory results, medical observations, or physiologic signals whose values or timestamps are related. Although the clinical relevance of relationships between these values or timestamps is well-recognized (1), common computer-based patient record systems have limited capability to query for patients based on these relationships, or to visualize these relationships in data displays. Improved support for identifying these relationships may help clinicians to make diagnoses, monitor disease progression, and evaluate therapeutic response in direct patient care; and may facilitate queries for patient populations in clinical research.

1.1 IDENTIFYING RELATIONSHIPS WITHIN TIME SEQUENCES OF CLINICAL DATA

Relationships between the values or timestamps of clinical data appear to be particularly important for interpreting quantitative clinical data (2). Figure 1 shows a time sequence of serum phenytoin results that needs correct interpretation. The patient is discharged between day 15 and 20 with a rising but not yet abnormal phenytoin level. This level will continue to rise unless the dose is decreased because phenytoin elimination follows zero-order kinetics in the therapeutic

range. Several weeks later the patient becomes symptomatic with phenytoin toxicity and is re-admitted for treatment. This situation, in which the patient is discharged with rising serum phenytoin and without a needed dose adjustment, appears to occur because an abnormal value did not yet exist when the patient was discharged, and because these data are usually presented to clinicians textually in a manner that makes trends difficult to see. While this example uses drug level data, similar considerations apply for other types of quantitative clinical data. Providing clinicians with support for identifying such relationships may reduce errors associated with the use of clinical data sequences (3).

Similarly, clinical research and quality assurance tasks may benefit from support for identifying populations of patients with specified relationships. Clinical research questions frequently have temporal aspects, for example, the duration of time over which a patient characteristic is true, or the characteristics of data within a time window relative to some clinical event (4, 5). Common clinical data retrieval systems do not support these types of queries (5-10).

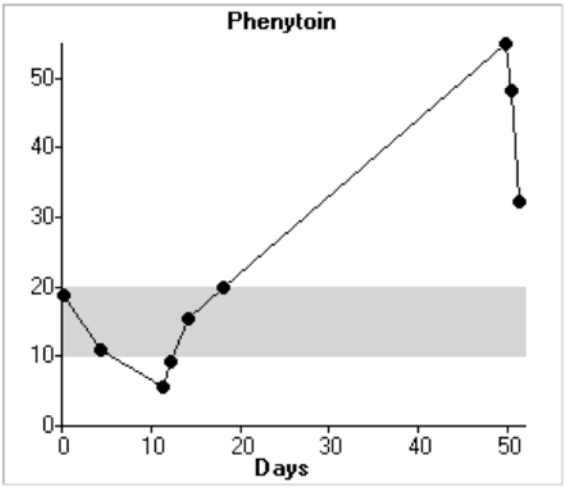


Figure 1: Serum phenytoin levels. Phenytoin's therapeutic range (10-20 mg/l) is shown in gray.

Furthermore, some conditions of interest are not represented in commonly used classification systems such as ICD-9 (11), and in those situations, the only way to distinguish patients of interest may be to identify characteristic relationships within time sequences of test results, observations, and clinical events (such as the phenytoin trend described above).

Temporal abstraction is a method for scanning time sequences of patient data for specified relationships between their values or timestamps (12). This approach defines an interval data type that represents a period of time over which a clinical state or process exists. Intervals have two time-stamps defining their endpoints. Specific states or processes are represented by abstractions, which correspond to data sequences satisfying specified constraints on the values of their data elements (mathematical patterns), or groups of intervals satisfying specified relationships between the timestamps of the intervals' endpoints (temporal patterns). Abstractions corresponding to mathematical patterns are usually defined in terms of a limited set of algorithms that define specific types of relationships between a data sequence's values, such as for detecting thresholds in a data sequence's values (*state* abstraction), or for identifying thresholds in a data sequence's slope (*trend* abstraction). Temporal abstraction has been successfully applied to identifying intervals in patient monitoring (13-20) and information visualization (21-23).

Though recent temporal abstraction systems provide additional algorithms to support basic statistical aggregation (24) or to detect characteristics in longitudinal data (25), they generally lack a framework for defining and managing specific algorithms designed for the analysis of a broad range of data types. The wide variety of clinical data in electronic medical records suggests that these systems should allow substantial flexibility in specifying the mathematical and temporal patterns that are used to define data sequences of interest. To be

broadly useful for identifying data sequences in individual patients, and retrieving patient populations containing those sequences, they should also provide interoperability with existing data stores and integration into standard networking environments. This work considers how temporal abstraction might be extended to better support these requirements.

1.2 CLINICAL DATA VISUALIZATION

Suboptimal data visualization in the medical record has been found to mislead clinicians (26) and lead to unnecessary testing (27), slower and less accurate searches for information (21, 22, 28), less timely decision-making (29), and wrong decisions (29, 30). The data management capability of information technology presents opportunities for improving the visualization of clinical data over how data is presented in the paper record, but electronic medical records have generally copied their form and organization from paper charts, in which data is typically presented textually and is organized according to administrative divisions of a hospital or clinic. While this strategy creates a tool that is familiar to users, in general it does not produce substantial productivity gains and may actually create systems that are more difficult to use (31).

The problem of displaying complex data sets in computer-based displays is not unique to medicine. Studies in aviation (32), hurricane tracking (33), industrial process control (34), and business (35, 36) have proposed displays that organize conceptually-related data elements in close proximity, use graphical forms to highlight important relationships between data elements, and emphasize critical data through sequencing. Evaluations have found that the techniques used by these displays have significant impact on the speed and accuracy of decision-making (32, 33).

Several experimental medical displays have incorporated similar techniques. Concept- or problem-oriented displays automatically link clinical problems and diagnoses with relevant data. These linkages support the presentation of data from multiple sources together in ways that facilitate detection of relationships between multiple data values. Displays organized by concept or problem may improve the speed and accuracy of data acquisition and interpretation (28). Graphical displays of clinical data have been found to help clinicians find and interpret data as well as facilitate timely and accurate decision-making (32-34). Altering the sequence with which data is presented may help physicians find important data faster (37). These studies suggest that there may be significant advantages to altering the form, organization, and sequence of traditional medical data displays.

In addition to altering a display's form, organization, and sequence, limited studies suggest that physicians may be able to find and interpret data more easily when a results display visualizes temporal patterns (21, 22). While these displays have been employed for visualizing small data sets in support of guideline-based care (38), it seems reasonable to consider whether similar techniques might be applicable to visualizing larger data sets for other patient care tasks such as inpatient results review, or for visualizing data of interest in quality assurance and clinical research.

1.3 PROCESS-ORIENTED DATA PROCESSING AND VISUALIZATION

This dissertation presents an integrated approach to improving the detection and visualization of disease and patient care processes for direct patient care, clinical research, and quality assurance. Key components of this approach are automated identification of temporal and mathematical

patterns, and the use of those patterns to control a display's form, organization, and sequence. This work has yielded software systems that implement each of these components: an automated interpretation program called PROTEMPA (Process-oriented Temporal Analysis), and a clinical data display program called TPOD (Temporal Process-oriented Display).

1.3.1 PROTEMPA (Process-oriented Temporal Analysis)

PROTEMPA is a data processing strategy and software library that allows specification and identification of mathematical and temporal patterns in time-stamped clinical data. Design considerations included a modular architecture, integration into standard networked computing environments, interoperability with existing clinical data stores, and scalability to large data sets. PROTEMPA extends existing temporal abstraction systems by supporting creation and maintenance of an extensible library of mathematical processing algorithms, called temporal abstraction primitives. These algorithms define general mathematical patterns applicable to time sequences of clinical data, and may be configurable with parameters constraining them to data sequences with particular characteristics.

1.3.2 TPOD (Temporal Process-oriented Display)

TPOD is a World Wide Web-based clinical data display that adapts its form, organization, and sequence to temporal and mathematical patterns in the data. It visualizes clinical data graphically and makes the corresponding numerical values available on demand. It explicitly visualizes intervals found by PROTEMPA and aggregates data that are related to found intervals, allowing users to quickly review the data that are relevant to a disease or patient care process. Existing electronic displays and paper charts may require significant manual searching in order to aggregate the same data.

1.4 EVALUATION

Two evaluations of PROTEMPA and TPOD were conducted. Goals were to determine whether PROTEMPA can detect and display disease and patient care processes of interest for patient care and clinical research tasks; and to determine whether TPOD improves the process and outcome of data interpretation and decision-making as compared with standard displays of clinical data. These evaluations focus on the processing and visualization of clinical laboratory data, but PROTEMPA and TPOD are designed to operate on any kind of clinical time series.

1.4.1 PROTEMPA for clinical research

The first evaluation aimed to determine whether PROTEMPA facilitates retrospective identification of patient populations containing temporal and mathematical patterns of interest to clinical researchers. A software program was written to identify and characterize cases of a severe form of pre-eclampsia called HELLP (Hemolytic anemia, Elevated Liver enzymes, and Low Platelets) syndrome (39) based on abstractions specifying patterns across multiple laboratory tests. This program invoked the PROTEMPA library with database connection information and abstraction specifications, and wrote the cases and intervals that PROTEMPA found to a text file. A script parsed the output file and categorized the cases into severity and outcome groups. The HELLP syndrome cases and outcomes classifications were validated by manual case review.

1.4.2 TPOD for patient care

The second evaluation aimed to determine whether a clinical data display based on TPOD facilitates the process and outcome of data interpretation and decision-making during inpatient results review. In this study, physician subjects reviewed and wrote orders on

unfamiliar patient cases in which laboratory data were presented using either a process-oriented display based on TPOD or a standard tabular numerical display. Subjects were asked to "think-aloud," and video screen captures were collected of subjects' interactions with the displays. The "think-aloud" transcripts were coded according to activities related to data acquisition, interpretation, and decision-making. A log file recorded the time required to complete the cases, the number of visits to the laboratory display, and the time spent using the laboratory display. In an exit interview, subjects were asked to describe their reactions to the two laboratory display styles.

1.5 CHAPTER GUIDE

Chapter 2.0 provides an overview of automated interpretation methods in medicine, descriptions of several approaches to visualizing quantitative data both in medicine and in other domains, and background on theories of quantitative data display. The chapter continues with an overview of methods for evaluating the effects of data displays on the process and outcome of interpretation and decision-making. Finally, it provides an overview of clinical laboratory data, how it is interpreted, and problems with how it is currently presented.

Chapters 3.0 through 5.0 report on the design, implementation, and evaluation of the process-oriented approach. Chapter 3.0 describes PROTEMPA's design and implementation. Chapter 4.0 describes an evaluation of PROTEMPA for identifying patient populations in clinical research. Chapter 5.0 describes TPOD's design and implementation, and an evaluation of PROTEMPA and TPOD in patient care.

Chapter 6.0 contains an overall discussion of the process-oriented approach, and suggests future work.

Appendix A contains a glossary that provides definitions of key terms for describing the methods used by TPOD and PROTEMPA, and definitions of medical terms used in this dissertation. Appendices B, C, D, and E contain documents related to the study in Chapter 5.0. Appendix F contains an example "think-aloud" transcript from the study in Chapter 5.0.

2.0 BACKGROUND

Computer-based patient record systems (CPRs) are used in a wide variety of activities, including patient care, clinical research, and quality assurance. The volume and scope of data being stored in modern CPRs is markedly increased as compared with early systems (40), and is likely to increase in the future as new techniques become available (e.g., genetic testing, biomarkers) (41). Existing tools for querying and reviewing patient data have not changed substantially, and are likely to contribute to clinician "information overload" (42). Two approaches for reducing information overload are automating the interpretation of clinical data, and improving clinical data displays. This chapter presents these two approaches as they apply to clinical data in general, and then discusses their importance in facilitating the interpretation of clinical laboratory results in particular.

2.1 AUTOMATED INTERPRETATION OF CLINICAL DATA

One strategy for reducing information overload is to provide clinicians with software systems that support interpreting and making decisions with clinical data, called clinical decision support systems, or CDSSs (43). The prototypical CDSS encodes clinical knowledge, and applies that knowledge to patient data in order to provide patient-specific recommendations (e.g., differential diagnoses, treatment options). Others are simply information management tools that provide data and knowledge for performing a task, but leave it up to clinicians to apply that knowledge to the

data. A third kind of decision support system focuses attention, such as data surveillance systems that alert clinicians when an abnormal test result has been reported or when a contraindicated drug has been ordered.

Decision-support systems have several components (see Figure 2). A *data source* provides access to a data store that is external to the CDSS. Early CDSSs often required the user to enter patient data into the system (i.e., the "external data store" was the user). A *knowledge base* provides storage and retrieval of clinical knowledge that has been elicited from domain experts. An *inference mechanism* encodes a set of algorithms that apply knowledge to data in order to make conclusions about those data. Finally, decision-support systems have a *mode of interaction*, which is the means by which users interact with the system. Some decision-support systems run continuously, performing periodic surveillance on a data source and communicating interpretations and/or decisions to users. Others scan a data source when prompted by a user.

Some CDSSs are designed to arrive at clinically meaningful conclusions by simulating the reasoning of expert clinicians, and thus are called expert systems. Expert systems have been successfully applied in experimental settings for diagnosis (44), management of infectious disease (45), intensive care unit monitoring (14, 46), guideline-based care (47, 48), computer assisted order entry (49), and clinical laboratory quality assurance (50, 51). Expert systems typically encode knowledge as sets of situation-action rules that detect changes in patient status and perform actions in response, such as displaying a message on a computer workstation's screen or sending an asynchronous message to a physician via pager (3, 52). Modern expert systems integrate patient data from multiple sources including laboratory information systems, medication administration records, and bedside monitoring devices (3).

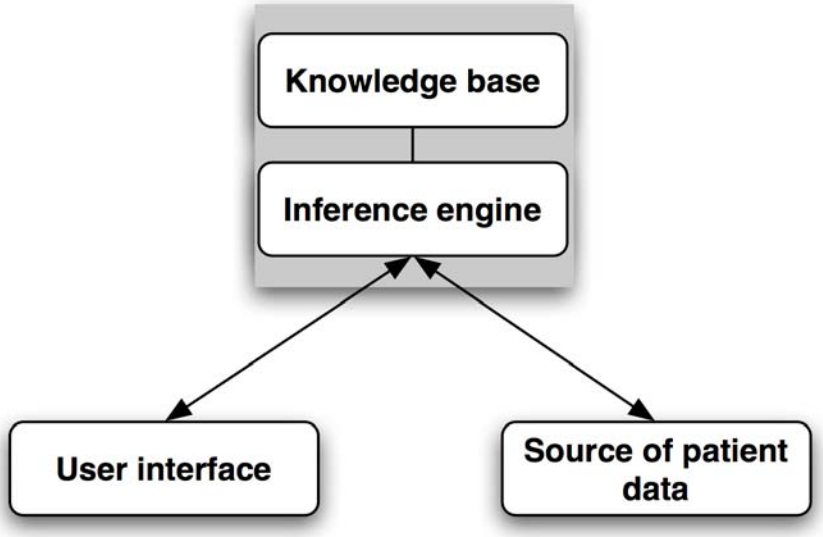


Figure 2: Clinical decision support system design.

2.1.1 Representing and reasoning with time

Early medical expert systems encapsulated the temporal aspects of clinical knowledge, such as the duration of a symptom, implicitly in symbols such as FEVER_LASTING_MORE_THAN_TWO_DAYS (53). Those expert systems could not use the temporal information in such symbols to make conclusions about what might be happening to the patient, nor could they specify temporal relationships between the finding described by a symbol and other data elements. Despite an implicit model of time, however, these systems performed very well for diagnostic tasks (54).

In contrast to diagnostic tasks, patient management tasks involve clinical reasoning that appears to rely more on the temporal sequence of clinical events and findings (54). Reasoning with quantitative data such as laboratory results often involves recognizing trends, identifying the frequencies with which tests are performed, and comparing contemporaneous results (2). In

order to identify such relationships in laboratory results and other kinds of data, expert systems are needed that can reason with the timestamps of these results.

To meet this need, recent expert systems in support of patient management can directly reason with clinical data's timestamps. These expert systems may employ a point-based model of time similar to that proposed by Vilain and Kautz (55). The point-based model has four main concepts: *time point*, *duration*, *interval*, and *relationship*.

A *time point* describes a clinical datum that is valid at a given timestamp, and is typically specified as a patient identifier, an attribute of the patient (e.g., a clinical observation), the value of the attribute, and the time-stamp at which the given attribute had that value, i.e., <John Doe, digoxin, 1.4, 4/2/2004 6:00am>.

Duration describes the temporal distance between two points (e.g., the distance between 4/2/2004 6:00am and 4/3/2004 7:30am is 1 day, 1 hour, 30 minutes).

A *time interval* describes a patient state or process that has occurred over a period of time. Intervals have been defined several ways in the literature. Shahar defined them as an ordered pair of time-stamps (56). Other point-based time models have defined them as a six-tuple specifying the earliest and latest possible starts of the interval, the earliest and latest possible finishes of the interval, and the interval's minimum and maximum possible durations (*minstart, maxstart, minfinish, maxfinish, mindur,* and *maxdur*) (57, 58). The latter definition allows for representing uncertainty in the beginning, end, and total time of a patient state or process.

A *relationship* describes the time distances between the endpoints of a pair of intervals. In the point-based model, relationships have been defined as an eight-tuple specifying the minimum and maximum distances between the start of the first interval and start of the second,

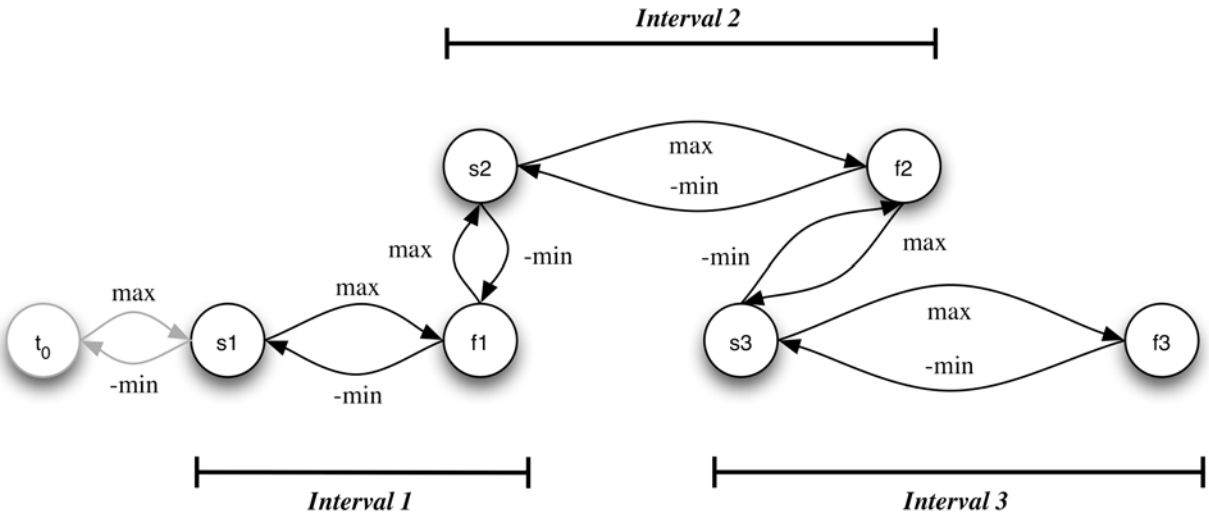


Figure 3: An example of a Temporal Constraint Network (TCN), as described in (59). A TCN is a directed graph in which nodes represent time points and arcs represent durations between them. Forward arcs represent the maximum duration between nodes, and reverse arcs represent the minimum duration between nodes. Intervals describing patient states may be represented as pairs of nodes, one for the start time (*s1*, *s2*, and *s3*) and the other for the finish time (e.g., *f1*, *f2*, and *f3*). A special time-zero node (t_0, in gray) may be used with distances between it and interval nodes (in black) to represent absolute start and finish times.

the start of the first interval and the finish of the second, the finish of the first interval and the start of the second, and the finish of the first interval and the finish of the second.

In the point-based approach, intervals and relationships are typically encoded as temporal constraint networks (TCNs, Figure 3, (59)). TCNs are directed graphs in which nodes represent the minimum and maximum starts and finishes of an interval, forward arcs represent the maximum amount of time between two nodes, and reverse arcs represent the minimum amount of time between two nodes. The values of unspecified arcs are determined by applying a shortest-path algorithm to the graph such as Bellman-Ford (60). Arcs between start and finish nodes of separate intervals specify relative temporal distances between those intervals. By adding a special time-zero node t_0 that is linked to an actual time, the graph can be anchored in real time, and arcs can be drawn between the time-zero node and interval nodes to specify

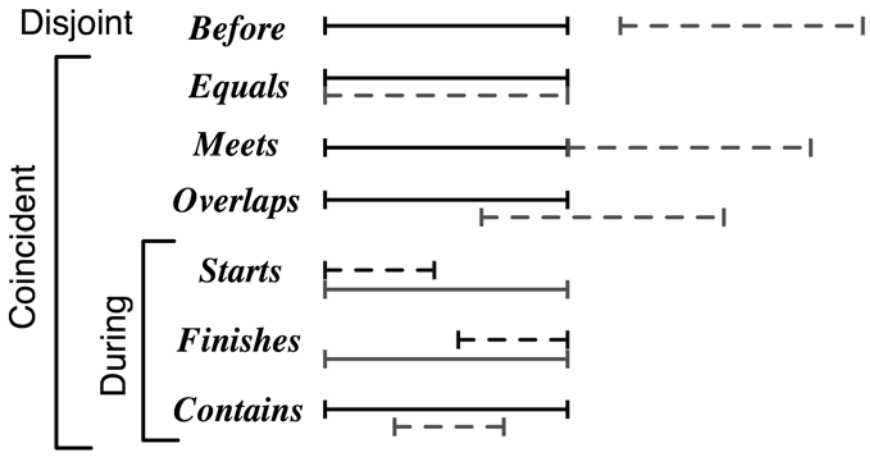


Figure 4: Temporal relations defined in Allen's temporal algebra (61). There are seven basic relations and their inverses (not shown). In the case of *Equals*, the basic and inverse relations are identical.

absolute starting and finishing times. In order to specify qualitative relationships within a TCN, two special arc values have been defined (58): epsilon (ε), the smallest possible nonzero temporal distance; and infinity (∞), the largest possible temporal distance.

The point-based model works well in clinical domains because specifying quantitative time distances between clinical events is possible. Determining if a time sequence is consistent with the constraints specified by a TCN's intervals and relationships has polynomial time complexity if at most one temporal relationship is specified between two endpoints of a pair of intervals (the "Simple" Temporal Problem, or STP described in (59)). An example TCN is shown in Figure 3.

An alternative model of time is Allen's temporal logic (61). Allen's logic has no time points, and instead of specifying relationships between intervals as durations between time points, it uses a set of qualitative temporal relationship descriptors (e.g., BEFORE, AFTER,

DURING), shown in Figure 4. Allen's logic was originally intended to represent time relationships in text documents, and thus supports significant ambiguity in the way these relationships are described. This high level of expressivity also makes the computation of relationships between events intractable in the worst-case (55). Furthermore, quantitative temporal relationships cannot be expressed in Allen's logic.

2.1.2 Temporal abstraction

The point-based time model is employed in one of the dominant strategies for automated interpretation of clinical data sequences, called *temporal abstraction*. Domains in which temporal abstraction has been evaluated include summarization of blood glucose monitoring data for diabetes management (16, 17), children's growth assessment (62), validation of high-frequency monitoring data in intensive care units (63), representing patient care plans (47), drug administration data for an expert system (64), clinical event monitoring (15), data mining of clinical databases (25), and evaluation of heart transplant patients (15).

Temporal abstraction distinguishes between two main types of abstractions: simple, and complex. Simple abstractions specify the abstraction of time-stamped data into intervals (the mathematical patterns introduced in Section 1.1); and complex abstractions specify the abstraction of groups of intervals with specified temporal relationships into higher-level intervals (the temporal patterns introduced in Section 1.1) (65). Temporal abstraction has been formally described by the Temporal-Abstraction Rules (TAR) language (66). The TAR language consists of rules and facts. Facts are time-stamped data values and intervals as described above. Rules "map" a fact to another fact at the same or higher level of abstraction. A rule is a statement of the form:

$$h(D,I,V) \leftarrow b_1(D,I_1,V_1),...,b_n(C,I_n,V_n) | \{c_1,...,c_m\},ifn,vfn$$

where h is the output fact, $b_1,\ldots,b_2$ are input facts, $c_1,\ldots,c_m$ are functions defining constraints on the values of the input facts, *ifn* is a function that computes the start and finish of h, and *vfn* is a function that computes the value of h. For each fact, D is a label, I is a time interval, and V is a value. In TAR, time-stamped data values are represented as intervals with the same start and finish. A knowledge base of TAR rules is a bottom-up deductive database (66). TAR has been used to prove that the temporal abstraction method always terminates (i.e., it never enters an infinite loop) (66).

A number of temporal abstraction systems have been reported for use in the clinical domain. The RESUME system is one of the first efforts at creating a temporal abstraction framework (56). RESUME supports a fixed set of temporal abstraction primitives: state, trend and rate, each of which may have one of a set of qualitative values (e.g., the trend may take the values INCREASING, DECREASING, or SAME). RASTA extends RESUME with a scalable distributed control algorithm in which each abstraction is evaluated in a separate process (13). RASTA has been implemented in a hypertension decision support system. ALMA (67) is a re-implementation of RESUME in the TAR language.

T-IDDM was a temporal abstraction system for summarizing patient blood glucose data (68). It introduced the concept of using temporal abstraction as part of a processing pipeline. This pipeline consisted of 1) a pre-processing stage in which data is transformed in some fashion, 2) a processing stage in which the transformed data is processed by temporal abstraction, and 3) a post-processing stage in which the intervals found by temporal abstraction are statistically analyzed. In T-IDDM's pre-processing stage, Fourier analysis was used to split blood glucose measurements into periodic and trend components, which were separately processed by temporal

abstraction (65). In the post-processing stage, statistical aggregation techniques were applied to calculate useful statistics such as the frequency of certain abstractions within a single patient (17). While T-IDDM's temporal abstraction implementation is not described in detail in the published literature, it appears to search for a more limited set of abstraction types than is available in RESUME and ALMA.

M-HTP, a system for abstracting parameters over time for comparing treatment of heart transplant patients to a clinical guideline, has many of the features of temporal abstraction, albeit implemented in a domain-specific fashion (15).

These temporal abstraction systems establish the method as a general-purpose tool for summarizing patient data through identifying mathematical and temporal relationships. Given the success of these systems, it appears that a limited pre-established set of mathematical patterns (e.g., state, trend) is sufficient for providing high-level descriptions of clinical data sequences for patient care tasks. However, clinical research queries and quality assurance studies may require the ability to extend the set of mathematical patterns (see Introduction). Furthermore, as these systems are designed to summarize patient data, they search for all possible abstractions in a dataset, not necessarily the subset of those required to answer a particular clinical query. Temporal abstraction has exponential worst-case time complexity (56) because the temporal relationship matching process evaluates all possible pairs of intervals to determine if they satisfy a specified relationship. However, complex abstractions typically specify only a few temporal relationships, so average case performance is sufficient for performing summarization of individual patients' data. Since temporal abstraction is parallelizable by patient, it can perform sufficiently even when summarizing multiple patients at a time.

Temporal abstraction is a general-purpose approach to identifying temporal and mathematical relationships in patient data. There are other approaches to analyzing time sequences of clinical data that have benefits and tradeoffs as compared with temporal abstraction that are described below. These systems typically address specific clinical reasoning tasks or have performance or pattern detection needs that are not met by the temporal abstraction systems described above.

2.1.3 Model-based methods

Kahn contended that no single method of representing time is sufficient for some domains (69), and implemented a system called TOPAZ for summarizing important patient-specific events in cancer chemotherapy treatment that combines two kinds of temporal data processing (18). The first method implements a physiological model representing a prototypical patient on chemotherapy, and abstracts raw data into intervals representing deviations from a normal therapeutic response. The second method scans for temporal relationships between the found intervals in order to construct a high-level overview of a case. A problem with using physiological models is that many diseases and therapeutic responses are not understood well enough to model them accurately, so a method that employs a model is not likely to be suitable for general-purpose use (53, 70). TOPAZ appears to be constructed specifically for providing decision support in cancer chemotherapy. However, the idea of combining mathematical processing of time-stamped data and high-level temporal reasoning is similar in concept to the more general-purpose approach described by this dissertation (see Chapter 3.0).

2.1.4 Trend templates

Another general purpose method of reasoning with time called trend templates (14) was developed in parallel with temporal abstraction. Trend templates denote a time-varying pattern in

multiple variables common to a diagnostic population. Predefined patterns of normal and abnormal trends represent disease processes as typical patterns of relevant parameters. In contrast with temporal abstraction, which outputs a single description of a patient's data, trend templates are organized into sets that represent alternative clinical states, and these sets compete for which one best fits the data.

Trend templates were implemented in TrenDx software (14). TrenDx is based on the Temporal-Utilities Package (TUP) (71), which implements a point-based model of time (55) using temporal constraint networks (59) to represent relations between time points and intervals. TrenDx has been applied to pediatric growth monitoring (72) and ICU data (73) and, in the case of pediatric grown monitoring data, reached the same interpretations of the data as a panel of experts, at a time no later than the experts (72).

While the trend template approach is suitable for analysis of a broad range of data sequences, it is optimized for monitoring tasks that may require analyzing frequently sampled parameters such as those found in the ICU. In doing so, the approach sacrifices flexibility. For example, trend templates cannot represent states that are derived from non-numerical data, such as drug administration records (e.g., "on digoxin"). Trend templates also cannot be organized as hierarchies in which templates contain "sub-templates." The latter feature would be useful for building sets of reusable templates as a way of facilitating knowledge engineering.

Trend templates are similar in concept to traditional clinical event monitor rules in that each template is a completely self-contained unit, and no state is preserved between templates. For example, if two templates specify the same threshold on serum potassium result values, both need to process the patient's potassium results. Trend templates have much more sophisticated handling of time, however, than the typical event monitor.

Despite several positive evaluations in which TrenDx was found to perform at or near the level of experts in pediatric growth monitoring (72), research related to trend templates has been mostly confined to that done by the original authors. One exception is the SENTINEL intelligent anesthesia monitor, which extends trend templates by using fuzzy logic to specify intervals of uncertain duration and variables of uncertain value (74).

2.1.5 Time series analysis

In the field of time series analysis (75), it is common to identify portions of a time series with patterns of interest by moving a window of defined length across the time series and analyzing the data in the window at any given time. Gall et al. provides a theoretical treatment of the need for a "moving" window when applying temporal reasoning to clinical databases, and enumerates a comprehensive list of parameters for controlling such a window's traversal of a time series (76).

Temporal Coupling Verification (TCV) is an implementation of a sliding window designed with stock market analysis in mind (77). TCV provides a mechanism for traversing a time series in a sliding window fashion to generate short subsequences called *segments* that are then evaluated for the presence of statistical patterns of interest. Statistical analysis functions are pre-defined and include *average*, *max*, and *min*. The basic functions may be combined to specify more complex patterns. This mechanism may be thought of as a statistical version of simple abstraction. TCV further defines a complex abstraction mechanism that searches for qualitative temporal relationships between pairs of statistical patterns. TCV is designed as an extension to Structured Query Language (SQL) (78), thus providing a powerful supplement to a standard relational database query language.

An advantage of TCV as compared with temporal abstraction is that TCV provides a broader range of pattern types, both because there are a wider variety of statistical functions available, and because the length of the sliding window can be specified. While temporal abstraction's built-in primitives define thresholds on single values or relationships between pairs of values, TCV can detect patterns in more than two consecutive values. For example, a function for finding increasing and decreasing trends could specify a sliding window length of three values. A drawback of TCV is its lack of support for specifying related groups of patterns, as are specified by the different possible values of a temporal abstraction STATE or TREND abstraction.

2.1.6 Segmentation

A signal processing technique called segmentation identifies inflection points in a time series in order to partition the time series into piecewise linear segments (79). There are two general kinds of segmentation algorithms: inductive and deductive.

Inductive segmentation statistically identifies segments of a time series with similar "shape" using clustering algorithms. Clustering partitions data such that similar data sequences belong to the same group and dissimilar data sequences belong to different groups. Similarity is measured by a distance function. Dynamic time warping (DTW) is one such method that was originally developed for speech recognition (80). DTW is a dynamic programming algorithm that attempts to match an input signal to templates that allow for variation in the lengths of components of the template. DTW is said to "warp" the time axis so that the distance between two time series becomes minimal with respect to a distance function. The cumulative value of the distance function yields a measure of similarity. Clusters correspond to basic abstractions in

the terminology of temporal abstraction, but DTW does not assign episodes with a semantically meaningful label.

As compared with trend templates and temporal abstraction, DTW may represent patterns more naturally in domains characterized by irregularly spaced data and where "normal" differs widely from patient to patient. For example, a DTW algorithm was found to detect acute rejection for a set of kidney transplant recipients with significantly higher accuracy than a group of experienced physicians (81). Acute rejection is detected from a patient's creatinine course, and normal creatinine for a kidney transplant recipient varies widely. DTW can only be used, however, in situations in which there are training sets of time series that are classified as normal or abnormal. In most patient situations, clinical laboratory data have consistent "normal" ranges from patient to patient, and training data are not available. Even if training data are available, there may be limited situations in which there is enough homogeneity in normal and abnormal patients for DTW to learn all of the relevant boundary conditions. Thus, while promising in specific restricted circumstances, DTW does not appear currently to be practical for use in a general patient population.

In contrast to inductive segmentation, deductive segmentation defines a set of shapes ahead of time, so segments have semantically meaningful labels. Frequently used shape descriptors include CONSTANT, LINEARLY INCREASING, and CONVEXLY DECREASING (79). These correspond to concepts identified by simple abstraction. Deductive segmentation is intended for analysis of high frequency data, and thus has potential application in intensive care unit data analysis. These algorithms aim to detect linear and non-linear curves in patient data with better performance than methods such as TrenDx (14). Deductive segmentation tends to be hampered by over-sensitivity to local maxima and minima, but may be

more robust than simple abstraction for outlier detection. Deductive segmentation has the potential to serve as input to a complex abstraction mechanism that could infer higher-level patterns (82), but there are no examples of such a system in the literature.

2.1.7 Probabilistic methods

Most of the systems described above, including temporal abstraction, are deterministic with respect to the concepts they infer from raw data. The clinical state or process represented by an interval is either present or absent. It may be possible to perform retrospective studies to calculate a positive predictive value for an individual abstraction's ability to identify a state or process of interest, but propagating such probabilities from simple abstractions to complex abstractions is likely to be problematic (83).

Nonetheless, in some domains, it is useful to reason with intervals whose "truth" during a period of time is uncertain. In TrenDx (Section 2.1.4), competing trend templates could be assigned numerical scores measuring of goodness of fit to the underlying data (14). The template that best describes the data is the template with the highest goodness of fit score. TrenDx's lack of support for sub-templates makes this feature feasible, because there is no need to propagate goodness-of-fit scores from lower to higher levels of abstraction.

Another approach is a probabilistic method based on Bayesian networks (84), called temporal belief networks (TBNs). TBNs replicate the structure of a belief network for multiple time points, and arcs between the same variable at different time points represent how the variable changes over time (2). Two problems are that the structure must be replicated at the smallest time unit that needs to be represented, and there is no support for different levels of abstraction.

An alternative temporal formulation of belief networks, called modifiable temporal belief networks (MTBNs), aimed to solve the drawbacks of TBNs by 1) providing a condensed representation of the network to facilitate construction and review and a mechanism for expanding the network to a full TBN upon deployment, 2) expressing temporal relations in multiple units, and 3) expressing relations between variables at more than one level of detail (85, 86). An MTBN for liver transplantation follow-up was implemented and found to have run-times significantly faster than traditional TBNs (85). There have been no evaluations of MTBNs in which clinicians directly interact with an MTBN-based system, so it is currently unknown what the implications of MTBNs are regarding user interactivity and performance. An open research question is how to visualize the uncertainty of found concepts, particularly when goals include reduction of information overload, focusing attention, and summarization.

2.1.8 Domain-specific systems

In addition to the general-purpose algorithms described above, several systems have been implemented that are specific to a single task but have features of interest. An electrocardiogram monitor was reported (20) in which multiple statistical techniques, trained by machine learning, detected different signal patterns in EKG tracings (e.g., P wave, QRS wave), and a temporal relationship detector scanned for combinations of these patterns that indicate cardiac abnormalities. In another report, software was implemented to detect long-term trends in laboratory test results for hepatitis (e.g., liver function tests, hepatitis antibodies) (87). The approach was similar to temporal abstraction, but instead of detecting states and trends in the data, the software detected changes of state (e.g., peak values, change from normal to elevated). These systems illustrate a need for integrating flexible classification strategies into general-purpose temporal abstraction systems.

2.1.9 Summary

Temporal abstraction is a general-purpose strategy for identifying temporal and mathematical relationships that has been successfully applied to several clinical tasks. The alternative approaches described above illustrate several potentially useful extensions, including 1) support for efficient pattern detection in frequently sampled data, 2) uncertainty, and 3) a broader array of mathematical pattern detectors. This dissertation addresses the third extension (see Chapter 3.0).

2.2 IMPROVING CLINICAL DATA VISUALIZATION WITH TEMPORAL ABSTRACTION

Another approach for reducing information overload may be to improve the presentation of clinical data. This approach involves understanding the cognitive principles behind how clinicians acquire clinical data from a display, and determining how different displays might facilitate or inhibit efficient data acquisition, interpretation, and decision-making. Common clinical data displays tend to adopt static paper display conventions. Given the volume and complexity of clinical data, it is likely that no one presentation is optimal for all kinds of data or for all tasks in which a kind of data is used. Displays may need to alter their presentations according to the meaning of the underlying data in order to maximize the efficiency of data acquisition, interpretation, and decision-making. This dissertation proposes the use of temporal abstraction (see above) to support such a display. In order to use temporal abstraction in this fashion, it is necessary to identify a mechanism by which a data display that can be altered in response to the discovery of temporal intervals.

2.2.1 Data display characteristics

A substantial body of work outside of the medical domain has identified a set of three component characteristics of data displays: the *form* of individual data elements, the *organization* of data elements into groups, and the *sequence* in which data elements appear (35). These characteristics appear to affect the nature of decisions made using a display in different ways (36).

In a study in which 60 MBA students evaluated sets of eight loan applications using a data display that described applicants on four relevant attributes, the data display's organization strongly affected the speed and accuracy with which subjects found data in the display, while form and sequence had a greater effect on data interpretation (36). In a simulated military pilot scenario, subjects integrated data for decision-making more effectively when four data elements were presented simultaneously on the screen than when the data were presented sequentially, one element per second (88). In another study in which subjects chose between alternatives that had multiple attributes, the order and style of presentation affected decision-makers' relative weighting of attributes (89).

Although studies evaluating clinical data presentation and decision-making are limited, their results are generally similar to those described above. A group of 315 clinicians who reviewed identical chief complaint, history, physical exam, and laboratory data in different sequences weighted information viewed late in the sequence more heavily and judged the probability of disease differently (37). Physicians' decisions to stop clinical trials early on the basis of negative effects were influenced by the form and organization of aggregate clinical response data (30). Displays that provided summary pages of data required for making decisions in an outpatient setting about a patient's microbiological or serum lipid status allowed physicians

to interpret data more rapidly with fewer errors than when using data organized by laboratory section (29, 90). Grouping data based on a clinical problem or concept tends to aggregate data required for a particular decision, and its benefit is consistent with the grouping effects noted previously with pilots and others (88, 91).

2.2.2 Data acquisition from clinical displays

Behaviors that are employed by physicians for data acquisition using paper charts appears to be complex and directly related to the organization of data on the page (92). Data acquisition behaviors include obtaining an overview, scanning for evidence of known problems, and identifying anomalous findings (93). Physicians may navigate to a page in the chart to satisfy one information need, but spontaneously decide to satisfy additional information needs while they are there. Furthermore, physicians might use others' accounts of a patient case in written notes rather than look at the "primary data" (e.g., laboratory test results) themselves, particularly for obtaining an overview. Given that electronic medical records typically use a similar basic organization as paper charts, one would expect similar results.

2.2.3 Graphical display of numerical data

Studies of textual numerical versus graphical data presentation suggest that graphics may support data interpretation and decision-making more effectively, particularly when quantitative comparisons must be made. In an evaluation of a two-dimensional graphical display in which the third dimension was expressed either numerically or graphically, airline pilots made more rapid decisions and used the third dimension more effectively when it was displayed graphically (32). In a hurricane tracking scenario, subjects made more rapid and better evacuation decisions when storm data was displayed graphically rather than numerically (33). In a process control study, fault conditions involving several variables were detected more effectively in graphical than in

purely numerical displays (34). Several studies have shown that the difference between numerical and graphical displays is most pronounced under time pressure (33, 94, 95). Similar results have been found in the medical domain. In an ICU setting, staff members correctly classified the respiratory status of patients in an average of 9 seconds using a metaphor graphics display (see below) compared to 18 seconds using a standard numerical flowsheet (96). These findings have been interpreted as evidence of "off-loading" some of the cognitive overhead of data evaluation to the visual system when graphics are used, allowing redirection of cognitive resources to the decision-making task (91, 95).

Given that a graphical display of quantitative data can outperform a numerical display, a question that arises is what kinds of graphics should be used. Graphical perception theory (97) attempts to answer this question. This theory views the graphical display of quantitative data as a form of data encoding that makes important patterns in multiple data values more explicit. Identifying these patterns requires the user to perform one or more perceptual tasks that different kinds of graphics support with varying degrees of accuracy. Commonly used perceptual tasks, ordered from most to least accurate, include:

1. Position along a common scale
2. Positions along nonaligned scales
3. Length, direction, angle
4. Area
5. Volume, curvature
6. Shading, color saturation

Using this theory, graphical display techniques can be classified from simpler to more complex according to the relative accuracy of the perceptual tasks that they require. Cartesian plots, for

example, rely on the tasks of decoding direction and position of points along a common scale. Pie charts rely on angle, area, and length, thus making them more challenging with respect to decoding quantitative data. One of the most complex kinds of graphics according to this theory is the metaphor graphic. A metaphor graphic is designed to evoke a corresponding object in the real world (96). Each feature of a metaphor graphic is supposed to represent a different data element, and different combinations of values of the component data elements produce different overall appearances of the graphic that ideally reflect states of the real-world object.

This theory does not propose that simpler graphics should be used at all times. Metaphor graphics can represent a lot of data in compact form. However, users must remember the meaning of each feature even if they use the data display only occasionally (98). Thus, training and experience may be an important factor in choosing a graphic. Clinical results displays have both everyday users and occasional users, and training on particular medical data displays is typically very limited. Therefore, given a choice between employing metaphor graphics that convey complex data in compact form and simpler graphics that employ more generic but familiar display techniques, theory suggests that simpler graphs may be more appropriate (99).

2.2.4 Visualization of temporal intervals

A typical way of displaying temporal intervals is as a horizontal bar with optional vertical lines demarcating defined start and finish time points, as shown in Figure 5. Several prototype systems have been described that employ this technique to visualize intervals identified by temporal abstraction.

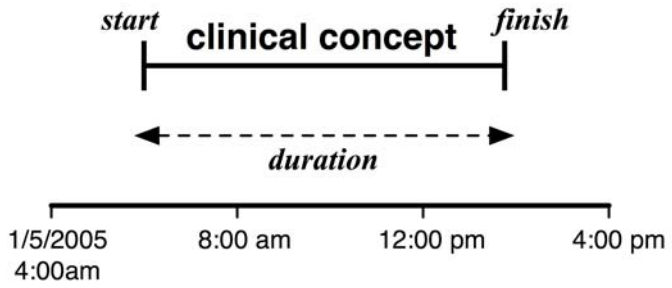


Figure 5: Graphical depiction of a patient state or process that occurs over an interval of time.

KNAVE was a tool for displaying automated summaries of a patient case, with the automated summaries created by temporal abstraction (22, 100). KNAVE was designed for displaying a limited data set related to a patient care plan. For example, a care plan for administering a chemotherapy protocol might monitor the patient's platelet and white blood counts, which together can be used to determine if the patient is experiencing a side effect of chemotherapy called myelotoxicity. The platelet counts, white blood counts, and episodes of myelotoxicity (if any) are displayed in timeline plots one above the other and are all aligned to the same horizontal time axis. Interactivity features include the ability to zoom into a period of time, display of frequency statistics regarding the fraction of time a concept was true about a patient, and the ability to perform sensitivity analysis by modifying a data value and seeing what effect it has on the derived concepts. The vertical size of each plot limits the amount of data that can be displayed on one screen, making it unsuitable for a general-purpose clinical results display.

A component of T-IDDM (Section 2.1.2) visualized clinically important episodes in a patient's blood glucose measurements alongside the raw data (68). In this case, intervals were directly superimposed on top of a plot of blood glucose data. The purpose of this display was

similar to that of KNAVE, to present a summary of the current status of a patient for a single patient care task.

A visualization tool for TrenDx (Section 2.1.4) called SmartDisplay was reported as being under development (23), but the tool does not appear to have been completed or deployed. SmartDisplay defined a protocol for communicating interval-based concepts between a temporal data analysis framework and the display that allowed it to serve as a front-end for any temporal data analysis system, including those that use temporal abstraction rather than trend templates. If TrenDx were driving SmartDisplay, data related to the highest-scoring template would be shown in the display along with time-line bars showing the intervals of data that have high similarity to the template. SmartDisplay appears intended for use as an ICU monitoring system, perhaps as a replacement to existing displays of physiological parameters that are usually available at ICU nursing stations (11).

A different class of temporal data displays aims to present a single-screen summary of patient status. The first example of such a system is Cousins' timeline display (101), which has the ability to display both interval-based and time-stamped patient data but no mechanism for deriving interval-based concepts. Lifelines (102) has similar features. The TimeLine system (103) appears only to support time-stamped data but displays informative icons for data items, such as a thumbnail of a chest x-ray that can be clicked to view a full-size image. Lifelines and TimeLine group data according to source-oriented categories, however TimeLine can display separate timelines on separate screens for each of the patient's clinical problems. These monolithic timeline displays may be of value as "consoles" for quickly ascertaining a high-level overview a patient. It is unclear, however, how to display the large volume of clinical data for a

typical patient, especially a hospitalized patient, without having to summarize some of the raw data. Temporal abstraction could serve as an automated summarization tool for this purpose.

2.2.5 Data display evaluation

Recent studies in medical informatics have explored the usability of medical information systems and their effects on clinical decision-making processes using techniques from the cognitive science literature (104). Cognitive science is concerned with the processes of perception, reasoning, problem solving and decision-making. The techniques used to explore these processes include videotaping user-software interactions; verbal protocol analysis, in which software users "think aloud" as they carry out tasks with software and their thoughts are classified using a coding scheme; and observing software users directly both in usability laboratories and in the field (105).

Published studies have primarily focused on the usability of information systems. These studies evaluate data entry tasks such as entry of coded data (106) and the effect of physician interaction with a computerized patient record on the patient interview (107, 108). Coding schemes are developed that classify the user's thoughts and intentions while performing tasks with the software and/or interacting with a patient, and users are asked to "think aloud" as they perform these tasks. Video is synchronized with recorded codes so that the user's actions, thoughts, and intentions can be correlated. Process measures are developed using the frequencies and distributions of codes during the time course of the study.

Cognitive science techniques have gained wide acceptance in software usability engineering (109) because they can provide measures of the success of a software design for a particular task, the degree of effort or "cognitive load" required to use the software, and the influence of the software on thought patterns and problem-solving strategies. These approaches

may also be helpful for observing the effects of alternative software designs on decision-making, although these types of studies have not been reported in medical systems.

Prior to the use of cognitive techniques, outcome measures and debriefing questionnaires were the primary sources of data for answering questions about usability and effects of a medical system on decision-making. Typical outcome measures include speed and accuracy of performing a task, and accuracy of making decisions using the information system. Process measures obtained from "think aloud" studies as described above can validate and explain the values of such measures. These studies may also bring to light unanticipated consequences of the information system, while outcome measures require that all consequences be anticipated (105). "Think aloud" studies are also believed to capture users' impressions of the system more accurately and in greater detail than questionnaires or exit interviews (110).

2.2.6 Summary

Further study is needed to determine the effects of form, organization, and sequence on data acquisition, interpretation, and decision-making. The data displays described above visualize limited sets of patient data, and their effects on data acquisition, interpretation, and decision-making were only evaluated with limited use cases. Nonetheless, these displays provide a validated set of clinical data visualization techniques to incorporate into process-oriented displays. Although techniques for visualizing temporal intervals have only been evaluated with small data sets, and their effects on clinical decision-making have not been evaluated at all, the results of these studies suggest that using temporal abstraction to control the display of clinical data may be a useful extension to existing experimental clinical display systems.

2.3 LABORATORY RESULTS PRESENTATION

Clinical laboratory tests provide data for a wide variety of clinical decision-making activities, including making diagnoses, monitoring disease progression, and measuring response to therapy (111). Laboratory tests are also a major data source for clinical research and medical process improvement studies. Laboratory data (both clinical and anatomic pathology data) comprised 94% of the objective data in one electronic medical record system (112), and it is commonly stated that 70% of patient data in a typical medical record originates from the laboratory (113). The volume of clinical laboratory testing has increased substantially in the past 30 years (114, 115), and is likely to continue increasing as genomic and proteomic analyses are incorporated into patient care (41, 116). Over 200 clinical laboratory tests are routinely performed at the University of Pittsburgh, and over 1000 are available.

Clinical laboratory data has unique features that complicate its interpretation as compared with other components of the medical record (99). Clinical laboratories communicate test results primarily through the medical record and usually without interpretive analysis. In contrast, other information in the medical record might be directly known to physicians (e.g., history, physical exam, progress notes), communicated verbally in addition to being documented in the record (e.g. specialist consultant reports, nursing data), or summarized in interpretive form (e.g., anatomic pathology, radiology, and consultant reports). Correct interpretation of laboratory results often requires identifying patterns in consecutive values of the same test and relationships between patterns in multiple tests. Yet, the standard tabular printout of laboratory results in the paper chart does little to make temporal changes clear. In inpatient settings, laboratory reports are typically organized by laboratory section (e.g. Chemistry, Immunology) and present results as numerical tables organized by name and time. The time axis is not proportional in these

displays, and test results related to a clinical problem may be spread across multiple sections. The spreadsheet-style views in most commercial electronic health records suffer analogous problems.

The majority of clinical laboratory tests performed for inpatients are for following disease progression and response to therapy rather than for making diagnoses (116, 117). Particularly when the patient's diagnosis is already known, clinical management may depend upon accurate "intermediate-level" (1) interpretations of time-related series of laboratory results. Such interpretations include patterns like abnormal and critical flags, delta checks, trends, and temporal relationships between patterns across multiple tests (e.g., simultaneously increasing blood urea nitrogen and creatinine during renal failure). Manually detecting clinically relevant patterns can be a time-consuming and error-prone process (1). Several information systems based on temporal abstraction (Section 2.1.2) have been developed for identifying these intermediate-level interpretations, and visualization software based on temporal abstraction have been successfully applied to interpreting and visualizing clinical laboratory data (Section 2.2).

Recent studies have identified incorrect interpretation of clinical laboratory data as a significant contributor to medical errors (118-120). In one study, "monitoring problems" including laboratory data were identified as a cause of 8% of physician ordering errors and the third most common cause of patient injury due to medications (118). In an analysis of therapeutic drug monitoring (TDM) in pediatric patients, 9% of 152 drug level tests were incorrectly used, contributing to medical errors (119). A study of 696 clinically important prescribing errors revealed that the most common factors associated with errors were changes in renal or hepatic function requiring a dose alteration (120). Indications of problems were found to be present in laboratory data but were not acted upon. These results suggest that existing techniques for

presenting laboratory data may be inadequate. Given the large volume of clinical laboratory tests, the need to identify relationships between multiple test results for correct interpretation, the utility of intermediate interpretations, and the relative lack of interpretive analysis in typical medical records, unique techniques may be needed to minimize the frequency of errors associated with using laboratory data.

Specific improvements have been proposed for laboratory data displays that may improve data acquisition, interpretation, and decision-making (121) with these data. Data acquisition may be improved by: grouping data by relatedness in time, organ system, physiologic system, or disease hypothesis; emphasizing key findings; eliminating redundancy and excess information; and reducing quantitative data to qualitative data. Characteristics that may facilitate data interpretation include rescaling and inclusion of derived values. Characteristics that may facilitate decision-making include optimizing the order in which data is presented, eliminating redundancy, and constructing data summaries.

One of the earliest systems reported in the literature to make use of these ideas was a touch-screen microcomputer system for sub-specialty consultants to find relevant laboratory results quickly (117). This system displayed test results of interest to a clinical subspecialty as a grid, or panel, of trend plots. Plots within a panel had time axes with the same start and stop dates to enable physicians to more easily see relationships across multiple tests. While this system was never formally evaluated, it was reported to have high acceptance among physician users (117).

The Clinician's Workstation (CWS) was a laboratory results display designed to improve utilization of laboratory tests in bone marrow transplantation units (27) through enhanced data organization. It provided a "console" of laboratory orders and results relevant to those units on a

single screen. Data were displayed numerically by default, although trend plots of one or more tests could be displayed on demand. The organization of tests on the screen appears to have been hard-coded. CWS was successfully deployed in pediatric bone marrow transplantation units, and after its first two years of deployment median charges for laboratory tests were significantly reduced (27).

The Query Clinical Information System (QCIS) used a clinical vocabulary to group laboratory data by concepts such as diagnoses (28). Concepts of interest were related by the vocabulary to sets of relevant laboratory, drug order, and other data. After a concept was selected from a search screen, the system presented a tabular display of laboratory data containing only the subset associated with the chosen concept. There was no capacity to associate test results with a concept only in the presence of certain test result values. QCIS was found to improve the speed and accuracy of data acquisition and interpretation compared to a source-oriented display (28).

A polygon-based laboratory data display was reported in which vertices represented individual tests, test values were mapped to a position on the line between the vertex and the center of the polygon, and the relative value of each test compared to its reference range was visualized against a background of concentric colored bands (122). The polygon served to provide a graphical display and to group results that would typically be interpreted together. No underlying knowledge base was ever reported for defining groups of data. The display was found to improve the speed and accuracy of interpretation of a set of abstract laboratory tests (123, 124). In contrast, two previous studies found that graphical display of the same laboratory tests had no benefit over tabular display when subjects were looking for single abnormal tests (123, 124), indicating a specific benefit of graphical displays when identification of patterns is

important. A drawback of this visualization tool is the lack of support for time sequences of laboratory results; the polygon can only display a snapshot of results that all occurred at the same time.

The Physician's Workstation (PWS) was an experimental outpatient electronic medical record that used a qualitative model of physiology to generate alerts related to laboratory test results (125). PWS displayed textual messages describing recent alerts in a text box adjacent to the results display (126, 127). Laboratory results were displayed graphically with trend plots, and tests' normal ranges were explicitly indicated so that physicians could detect abnormal test results at a glance. In addition to displaying alerts and trends, PWS facilitated interpretation of laboratory data by providing a selectable list of the patient's current diagnoses, and by highlighting data associated with the currently selected diagnosis. The data associated with a diagnosis was dynamic such that laboratory test results could be highlighted only if an abnormal value or trend was present. It is possible that dynamic highlighting gave PWS's graphical display added value, although no evaluations of these techniques were ever published. Nonetheless, the PWS nonetheless is an example of how automated interpretation, conceptual grouping, and graphical display might be used to highlight important patient conditions.

These systems provide evidence that improved organization and graphical display of laboratory data may facilitate earlier and more accurate detection of clinical problems by clinicians, although none have addressed the use of laboratory results for clinical research and quality assurance activities. The PWS's integration of graphical display and dynamic data organization seems promising, and it may be useful to evaluate the effects of these techniques on data interpretation and decision-making.

3.0 PROTEMPA: PROCESS-ORIENTED TEMPORAL ANALYSIS

Chapter 1.0 introduced an integrated approach to improving the detection and visualization of disease and patient care processes that are reflected by temporal and mathematical relationships in time sequences of clinical data. This chapter presents the design and implementation of the temporal abstraction component of this approach.

3.1 INTRODUCTION

Clinically important processes in individual patients may be recognized by physicians through temporal patterns in data collected from the history, physical examination, and diagnostic tests (128). Understanding these patterns and processes is crucial to making correct diagnoses, determining treatment, monitoring or predicting response to therapy, and following or predicting disease progression. Detecting these processes accurately often requires physicians to identify relations between the values and times of multiple data elements within and among data types (1, 65, 129). Information systems have been designed to assist the physician in performing this interpretation task through the method of *temporal abstraction* (14, 15, 20, 56, 63, 65), which may be defined as the process of inferring high-level interpretations from time-stamped data (see Introduction and Background).

Temporal abstraction may be viewed as a two-stage process. The first stage, *low-level abstraction*, is the identification of temporal intervals in raw patient data. The second stage,

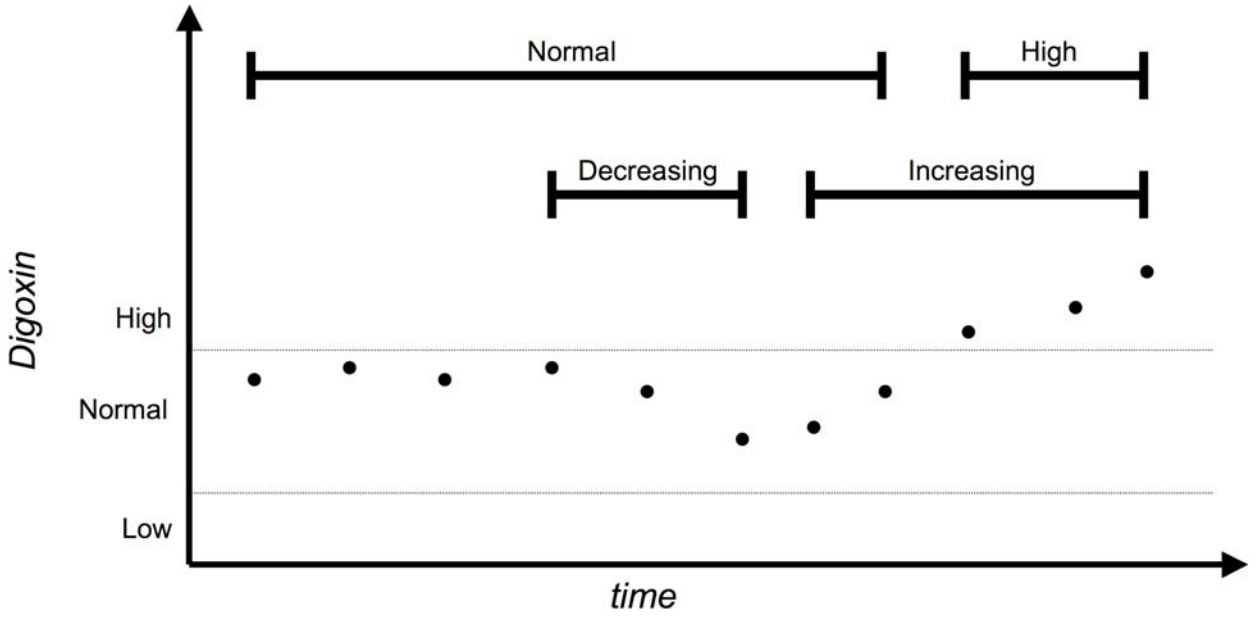


Figure 6: Temporal abstraction of digoxin drug levels. Temporal abstraction is the process of inferring high-level concepts from time sequences of raw data. In the example depicted in this figure, a patient's serum digoxin measurements are used to infer periods of decreasing values and increasing values. A period of normal levels can be inferred from the first eight values because they are all within the normal range (between the thin horizontal lines). The last three values are above the upper limit of normal and thus indicate that the patient had toxic levels during that time period.

high-level abstraction, is the process of combining groups of intervals that satisfy defined temporal relationships into higher-level abstractions. These stages are analogous to the *simple abstraction* and *complex abstraction* mechanisms described by Bellazzi et al. (65).

Low-level abstraction may employ a variety of heuristic, statistical or model-based pattern recognition strategies, techniques from the field of time series analysis (75), or classifications (*state, trend, gradient,* and *rate*) (56, 65, 130). The optimal techniques for a given data type or pattern within a data type are not always predictable and may need to be determined by experimentation. Figure 6 shows several examples of low-level abstractions inferred from a series of digoxin drug levels.

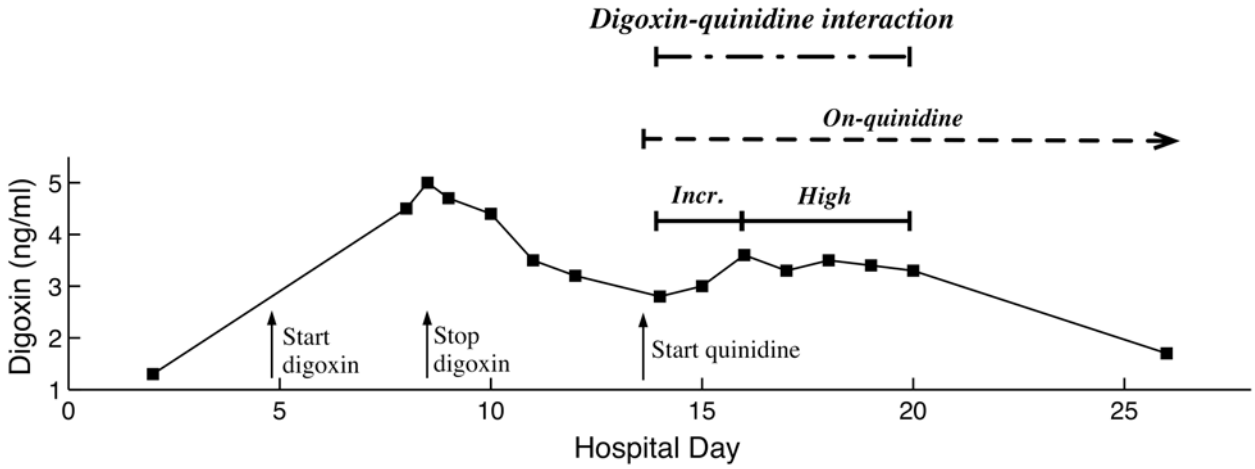


Figure 7: Temporal abstraction of serum digoxin drug levels and quinidine administration data. Periods of increasing digoxin (labeled *Incr.*), high digoxin (labeled *High*), and *On-quinidine* are inferred through the process of simple abstraction. From this information, the complex abstraction *Digoxin-quinidine interaction* may be inferred.

In contrast, effective high-level abstraction does not require heterogeneous techniques: a typical strategy is to define a formalism for describing constraints on the temporal relationships between pairs of intervals and a mechanism for determining whether a set of intervals satisfies those constraints (59, 61). An example of a high-level abstraction in the clinical laboratory domain is shown in Figure 7. A patient with renal failure was treated with digoxin beginning on hospital day 5, and serum levels of digoxin quickly rose above the therapeutic range (0.8-2.2 ng/ml). Digoxin therapy was subsequently discontinued, and the digoxin level declined until about day 13, when it began to rise without additional digoxin treatment. A likely explanation is that the patient was started on quinidine, a drug that decreases digoxin's volume of distribution and renal clearance. Thus, in the absence of concurrent digoxin therapy, intervals of rising followed by high digoxin blood levels early in a quinidine treatment interval could be abstracted to an interval of quinidine-digoxin interaction.

This chapter presents a novel software system that aims to provide flexible low-level abstraction and general-purpose high-level abstraction called Problem-Oriented Temporal

Analysis (PROTEMPA). An advantage of PROTEMPA as compared with existing temporal abstraction systems is that it provides a framework for defining, storing, executing, and sharing a library of general-purpose or task-specific algorithms, called *temporal abstraction primitives*, which employ arbitrary mathematical processing algorithms. These primitives may define general mathematical patterns within time-stamped data sequences and are configurable with parameters constraining them to sequences with particular characteristics. This framework makes it easy to develop, test and deploy heterogeneous algorithms optimized for particular tasks, and to aggregate found intervals across these algorithms into complex abstractions. PROTEMPA may be used either in a data mining (retrospective) mode to identify temporal patterns in large data sets, or in an event monitoring (prospective) mode to identify patterns of interest as they develop.

3.2 DESIGN

PROTEMPA is an object-oriented software library with a modular architecture that is callable through defined Application Programming Interfaces (APIs). It has four modules, shown in Figure 8, that provide 1) a framework for defining temporal abstraction primitives and processing data with those primitives (the Algorithm Source), 2) a framework for specifying algorithm parameters and interval relationships that define abstractions of interest (the Knowledge Source), 3) a connection to an existing data store (the Data Source), and 4) a data processing environment for finding abstractions in time-stamped data (the Abstraction Finder). The first three modules have back ends that implement environment- or application-specific features.

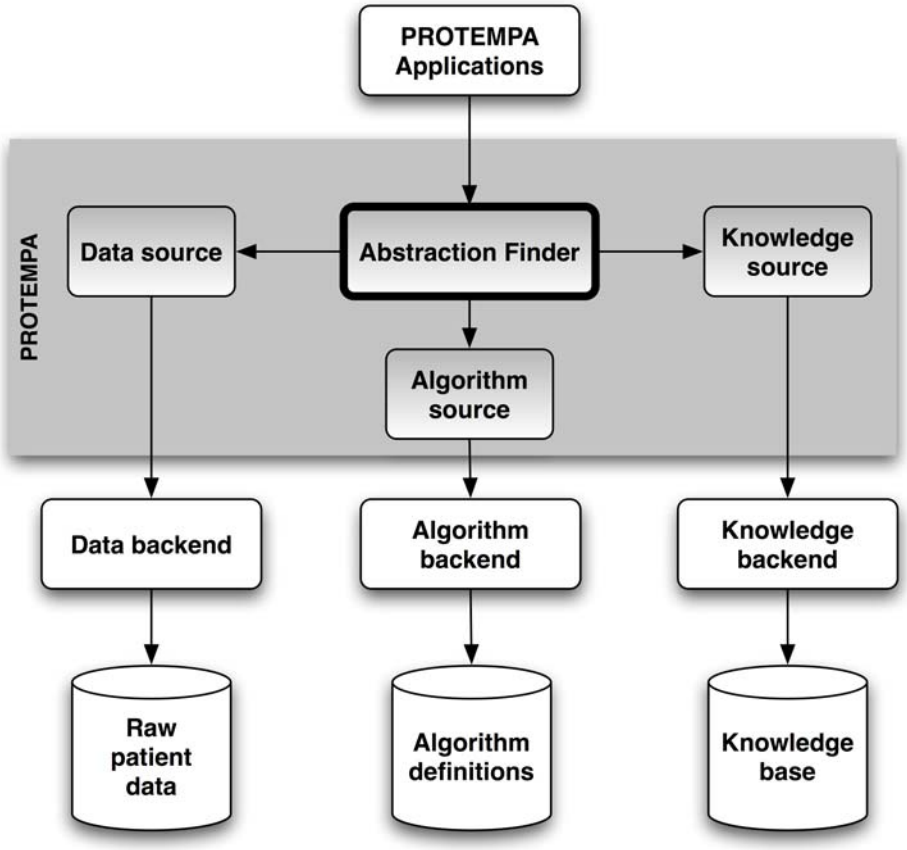


Figure 8: PROTEMPA architecture. PROTEMPA is a modular software library that implements an extension of the temporal abstraction method (56) for retrospective clinical data retrieval. The Abstraction Finder controls data processing and is supported by the Data Source, Knowledge Source, and Algorithm Source modules. Arrows represent dependencies between modules. A PROTEMPA application is a program that calls the PROTEMPA library through defined APIs.

3.2.1 Data model

PROTEMPA provides a set of classes for representing data values and computed intervals, shown in Figure 9. PROTEMPA represents information about a patient as a time sequence of *parameters*. Two types of parameters represent patient data: the time-stamped data value, and the constant data value. Time-stamped data values represent any kind of data that has a timestamp, and are stored with a temporal granularity unit (*tstampUnits* in Figure 9) that reflects how precisely the value's timestamp is known. Constant data values represent data that do not have a timestamp (e.g., the patient's gender). A third type of parameter, the interval, represents interpretations of patient data that have been computed by PROTEMPA. PROTEMPA's interval representation is based on a point-based model of time (see Background), and thus intervals have a starting time and a finishing time. When intervals are computed, their starting and finishing temporal granularity units (*startUnits* and *finishUnits* in Figure 9) are assigned the highest possible level of precision, which corresponds to the lowest level of granularity used in the computation.

3.2.2 Abstraction mechanisms

A program using PROTEMPA calls an API provided by the Abstraction Finder to scan patient data for one or more abstractions over a defined date range. The Abstraction Finder retrieves definitions of each abstraction from the Knowledge Source. It then extracts the data needed to compute those abstractions from the Data Source, performs temporal abstraction (calling the Algorithm Source as required), and returns found intervals to the calling program. The sequence of interactions between PROTEMPA's modules in response to a call to the AbstractionFinder API is shown in Figure 10.

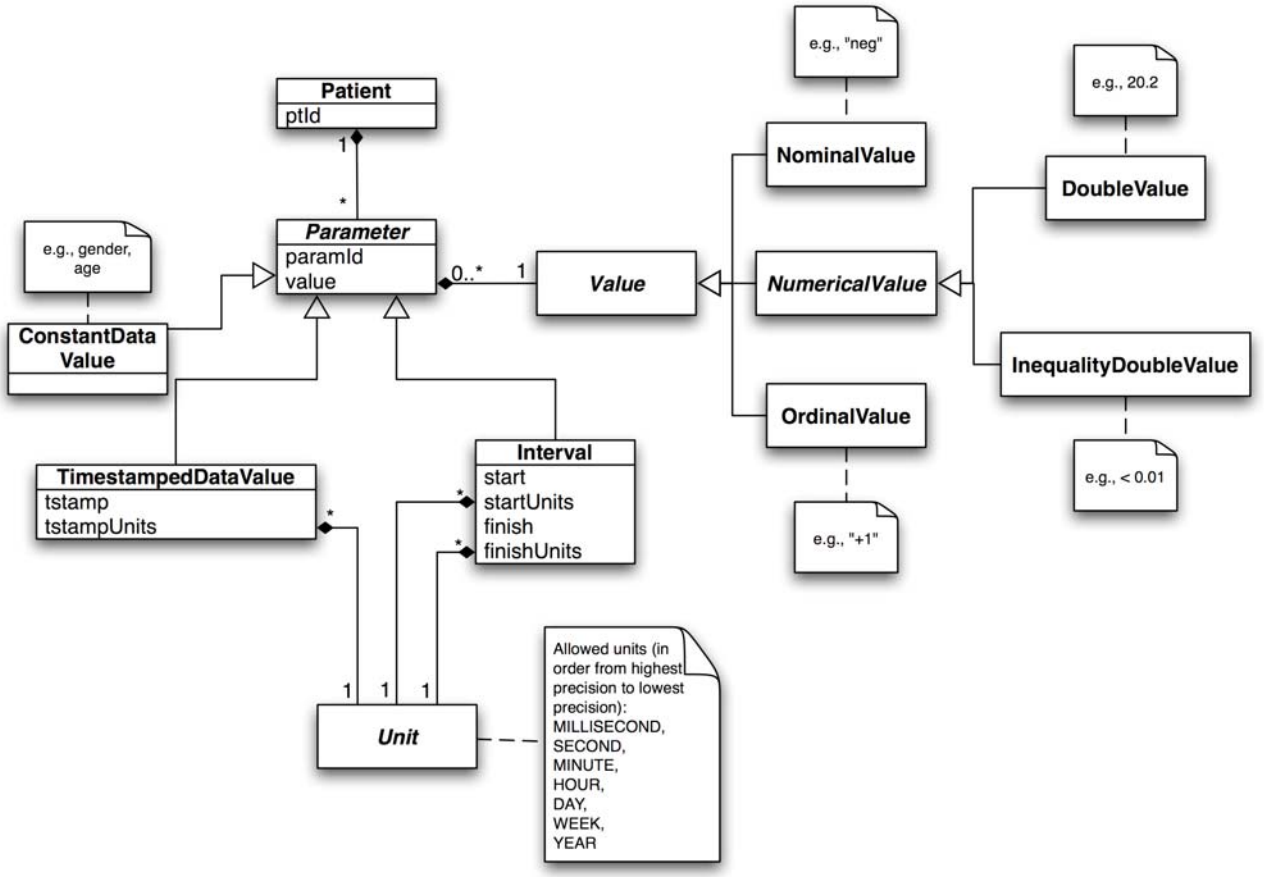


Figure 9: Class diagram of PROTEMPA's data model. PROTEMPA represents information about a patient as a time sequence of *parameters*. Parameters have a unique identifier, and optionally have a value at one of three levels of measurement. This class hierarchy models three types of values representing different levels of measurement: *NominalValue* (categorical without ordering); *OrdinalValue* (categorical with order such as "mild, moderate, or severe," or semi-quantitative +RANGE scoring such as +1 to +5); and *NumericalValue*, which encompasses both interval (numerical with arbitrary zero point) and ratio (numerical with true zero point). The numerical type is subdivided into *DoubleValue*, which encompasses double-precision values, and *InequalityDoubleValue*, which encompasses numerical values prefixed by an inequality operator such as "< 10." The latter is a common method of reporting a laboratory result above or below the range that can be accurately reported for a given test.

PROTEMPA's Abstraction Finder decomposes temporal abstraction into three mechanisms: a low-level mechanism that applies algorithms defined in the Algorithm Source to time-stamped data, and two high-level mechanisms (discussed below) that apply interval relationships defined in the Knowledge Source to previously-identified temporal intervals. The low-level mechanism scans a sequence of time-stamped data using a sliding window mechanism based on Temporal Coupling Verification (TCV, see Background) to select successive data subsequences, as shown in Figure 11. These subsequences are passed to the Algorithm Source for processing. The width of the sliding window, identity of the algorithm, and specific arguments for the algorithm are defined in the Knowledge Source. When a data subsequence matches the mathematical constraints specified by an algorithm and argument set, the low-level mechanism returns a named time interval with start and finish times and temporal granularity units corresponding to the timestamps and units of the earliest and latest data values in the matching subsequence.

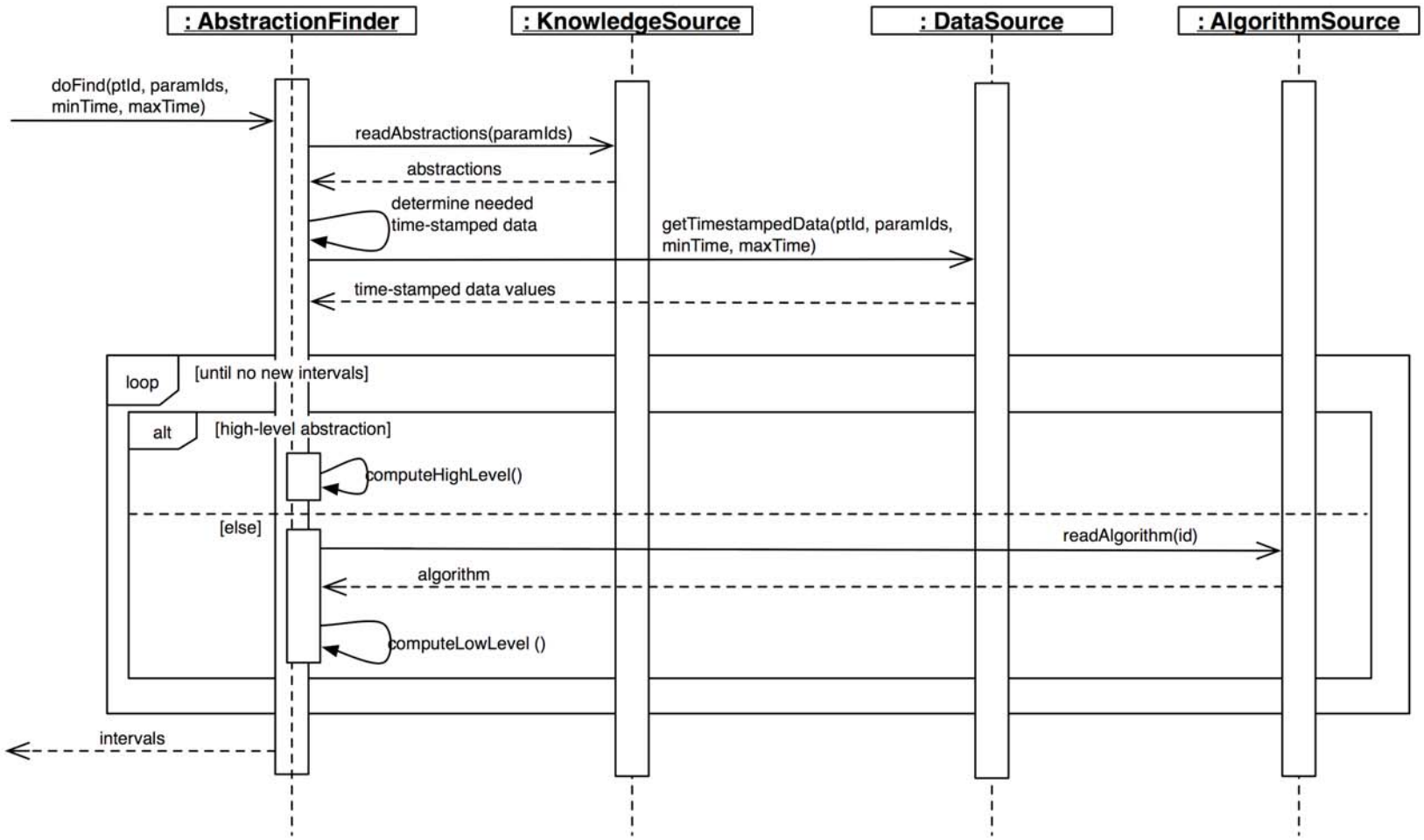


Figure 10: Sequence diagram of a PROTEMPA session. A program calls the Abstraction Finder's *doFind* method with a patient id (*ptId*), a set of abstract parameter ids (*paramIds*), and optional minimum and maximum times. The abstract parameter definitions corresponding to the *paramIds* are retrieved from the Knowledge Source. Then, the Data Source's *getTimestampedData* method is called to retrieve the time-stamped data. The minimum and maximum times are passed on to the *getTimestampedData* method, which returns only data with timestamps within the given minimum and maximum times. Intervals are computed and returned to the calling program. *paramIds* = a set of abstract parameter ids.

Processing for the high-level abstraction mechanisms is contained within the Abstraction Finder module. The first mechanism, *temporal pattern*, scans for groups of intervals with sequential, overlap, or co-occurrence temporal relationships. Relationships are specified in the Knowledge Source as minimum and maximum temporal distances between the endpoints of pairs of intervals, and constraints may also be specified in the Knowledge Source on the minimum and maximum duration of each interval in the group. These relationships and constraints are similar to those described by temporal constraint networks (TCNs, see Background and (59)). When a group of intervals are found that are consistent with the defined relationships and constraints, a named interval is created that typically encompasses the temporal extent of the group but may alternatively be temporally offset relative to one of the intervals in the group. The second high-level mechanism, *temporal slice*, processes all intervals of a given type as a chronological list and returns new intervals that are copies of an ordinal range, or "slice," of the intervals in the list. This mechanism can return the first, last or other intervals of an abstraction based on arguments specified in the Knowledge Source. The high-level mechanisms are similar to the temporal pattern matching (56) and cardinality constraint (24) mechanisms described by Shahar and coworkers.

The Abstraction Finder also implements an *interval combination procedure* that joins pairs of intervals of the same type if their nearest endpoints are within a defined maximum time limit, similarly to the previously described horizontal temporal inference and temporal interpolation mechanisms (56). When the limit is satisfied, the pair of intervals is replaced by a new interval of the same type that spans the two intervals' temporal extent. These optional combination limits are specified for the low-level and temporal pattern mechanisms as part of the definition of an abstraction in the Knowledge Source.

The Abstraction Finder module implements a data processing sequence that incorporates the three abstraction mechanisms (Figure 12). When processing is initiated, the low-level mechanism scans time-stamped data and identifies all intervals corresponding to low-level abstractions. The temporal pattern and slice mechanisms then operate on intervals found by the low-level mechanism, repeatedly scanning for new intervals until no more can be found. As the Abstraction Finder identifies new intervals, they become available for further processing. Intervals are cached so that subsequent calls to the Abstraction Finder's API do not cause the same intervals to be re-computed.

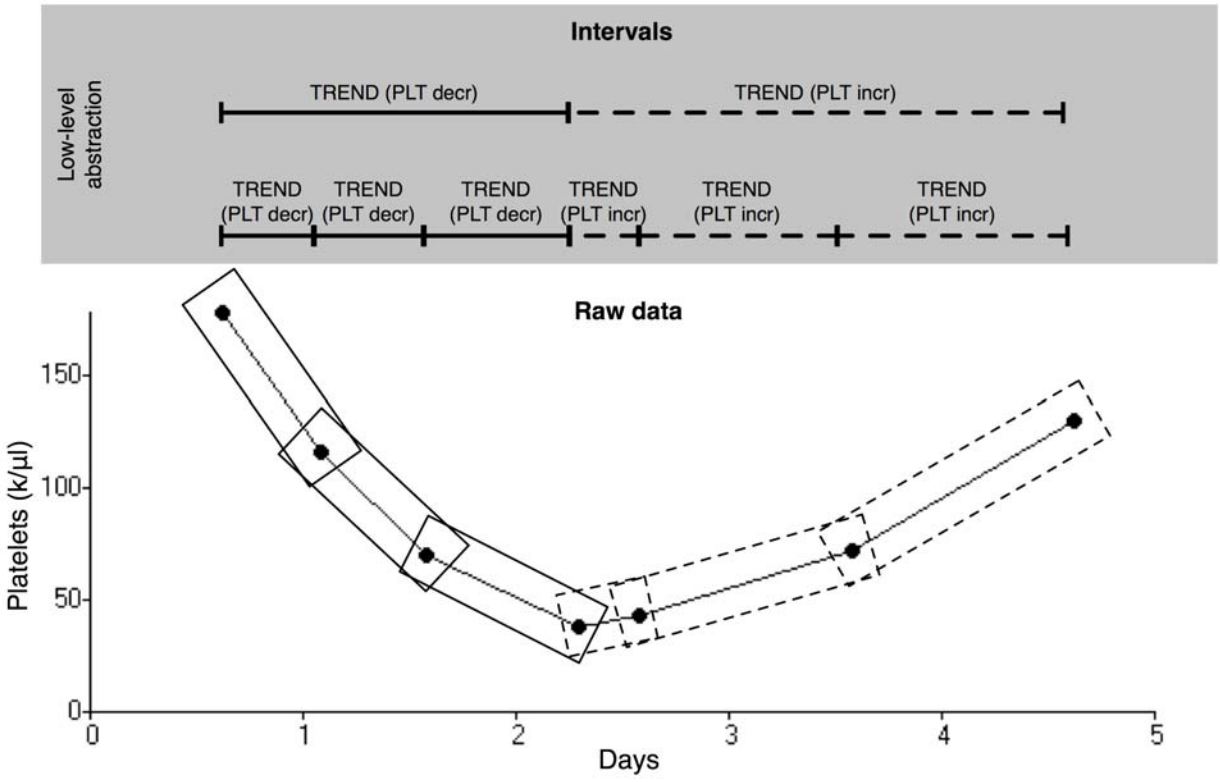


Figure 11: Illustration of the low-level abstraction mechanism scanning a time series of platelet counts. In this example a trend primitive (*TREND*) is used with criteria for determining if adjacent platelet values are increasing (*PLT incr*) or decreasing (*PLT decr*). After the intervals are identified, the interval combination procedure (see Design) combines adjacent *TREND (PLT incr)* intervals and adjacent *TREND (PLT decr)* intervals.

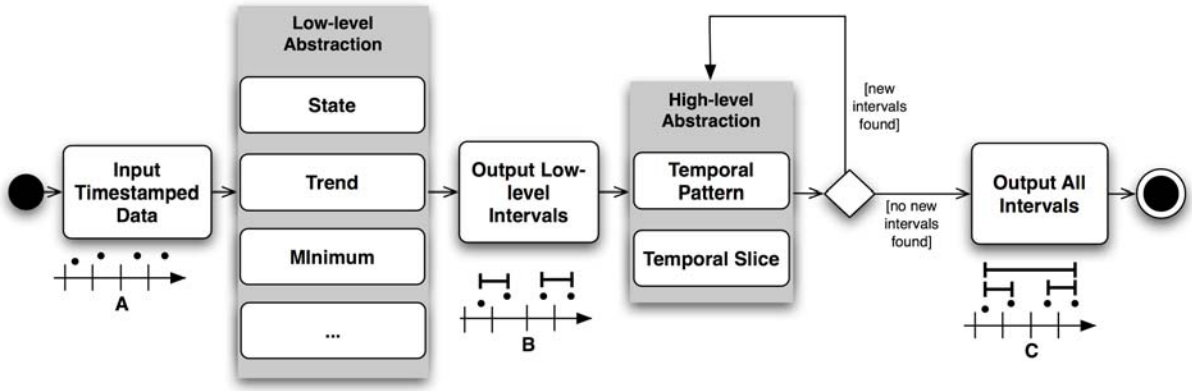


Figure 12: Activity diagram of PROTEMPA's processing sequence. The low-level abstraction mechanism scans time-stamped data sequences (A) for time intervals that correspond to low-level abstractions defined in the Knowledge Source (Figure 8). Found intervals (B) are subsequently scanned by the high-level mechanisms, which add new intervals corresponding to defined temporal pattern and temporal slice abstractions (C), repeatedly processing all intervals until no more are found.

The Abstraction Finder is supported by the Algorithm Source, Knowledge Source and Data Source modules. The Algorithm Source provides a method for storing, and a run-time environment for executing, temporal abstraction primitives that support the low-level temporal abstraction mechanism described above (class diagram shown in Figure 13). The primitives are implemented as algorithms in an arbitrary programming language. Algorithms may be written to identify a specific data pattern and be essentially self-contained, or they may describe a general pattern (e.g., the *state* and *trend* classifications from previous temporal abstraction systems) and be passed arguments (starting or ending cutoff values, slope, etc.) that define particular instances of the pattern for different settings. When arguments are used, they are specified in the Knowledge Source and passed to the Algorithm Source when the algorithm is called. The Algorithm Source backend provides the algorithm storage and run-time environment.

The Knowledge Source provides for storage and retrieval of abstractions (class diagram shown in Figure 14), each of which specifies an abstraction mechanism to use and a set of mechanism-specific parameters. Abstractions that use the low-level mechanism specify the name of a temporal abstraction primitive from the Algorithm Source, a set of constraints including duration and sequence length limits for the sliding window, and, if appropriate, arguments for the primitive's parameters. Abstractions that use the temporal pattern mechanism specify a set of temporal relationships between a group of previously defined abstractions and, if appropriate, constraints on the durations of the abstractions in the group (Figure 15). Abstractions that use the temporal slice mechanism specify a previously defined abstraction and an ordinal range (Figure 16). Abstractions that use the low-level or temporal pattern mechanisms also optionally specify interval combination limits for use by the interval combination procedure. A Knowledge Source backend connects to a knowledge base for storage.

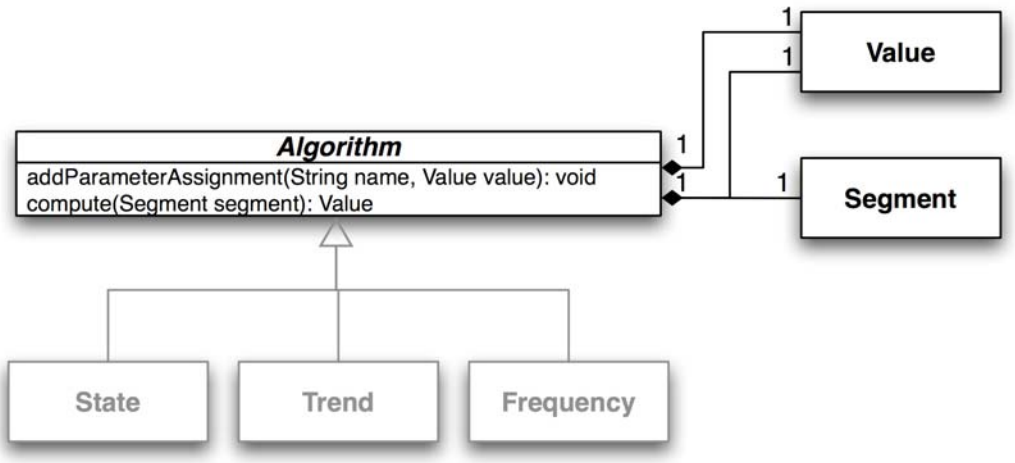


Figure 13: Class diagram of the Algorithm Source's algorithm model. When the Algorithm Source's *readAlgorithm* method is called (Figure 10), algorithms are returned as *Algorithm* objects. Values for an algorithm's parameters are set via an *Algorithm* object's *addParameterAssignment* method, and data subsequences are sent to an algorithm for processing as *Segment* objects via an *Algorithm* object's *compute* method. The Algorithm Source backend defines concrete implementations of *Algorithm*.

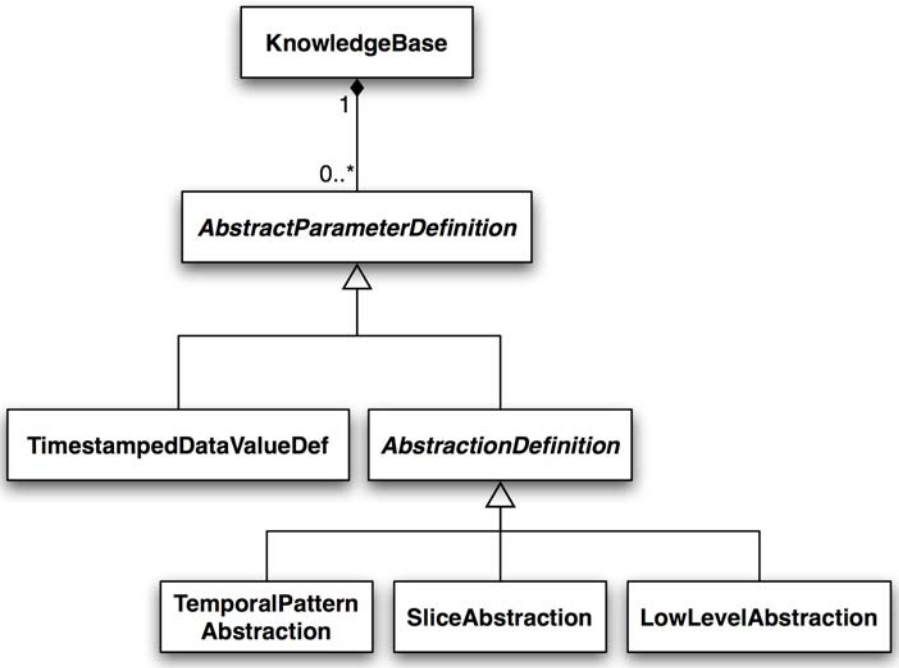


Figure 14: Class diagram of PROTEMPA's knowledge model. Data type definitions are represented by *TimestampedDataValueDef* objects, and abstractions are represented by *AbstractionDefinition* objects. Subclasses of *AbstractionDefinition* implement the three abstraction mechanisms. When the Knowledge Source's *readAbstractions* method is called (Figure 10), abstractions are returned as *TemporalPatternAbstraction*, *SliceAbstraction*, and *LowLevelAbstraction* objects.

The Data Source provides a connection to a physical database containing time-stamped clinical data, and a mapping between the terminology of the database and the terminology used in the Knowledge Source (130). The database connection and mapping are implemented in the Data Source backend. The Data Source provides data to PROTEMPA as instances of the *TimestampedDataValue* and *ConstantDataValue* classes shown in Figure 9.

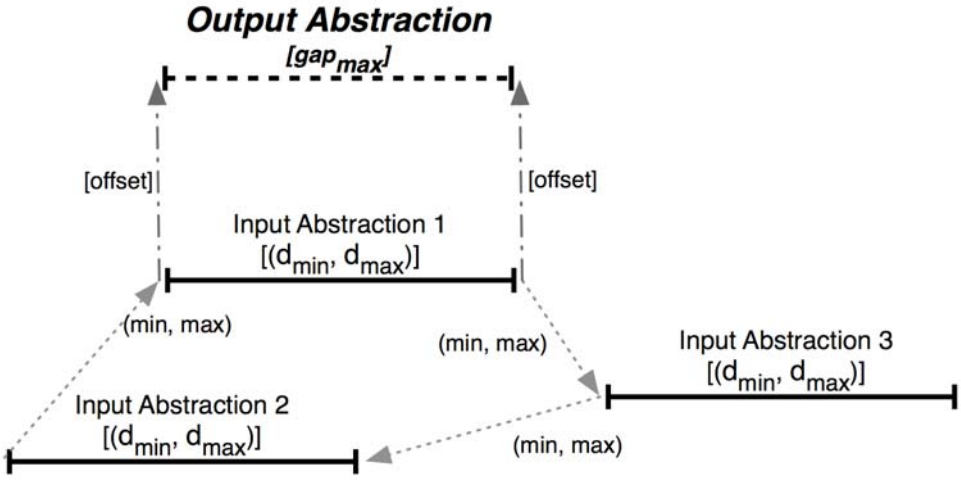


Figure 15: Temporal pattern abstraction definition. Abstractions that use the temporal pattern mechanism specify a group of abstractions (Input Abstractions 1 to 3), and temporal relationships (gray dotted arrows). Relationships are specified as minimum and maximum distances (min, max) between the endpoints of pairs of the input abstractions. For example, (0, ∞) indicates that the second time point in the relationship must occur on or after the first point (∞ = largest possible time distance). Constraints may be specified on each input abstraction's minimum and maximum duration (d_{min}, d_{max}), offsets of the endpoints of the output abstraction from the endpoints of the input abstractions (gray dashed arrows), and the interval combination limit (gap_{max}) for the output abstraction. Constraints in square brackets are optional.

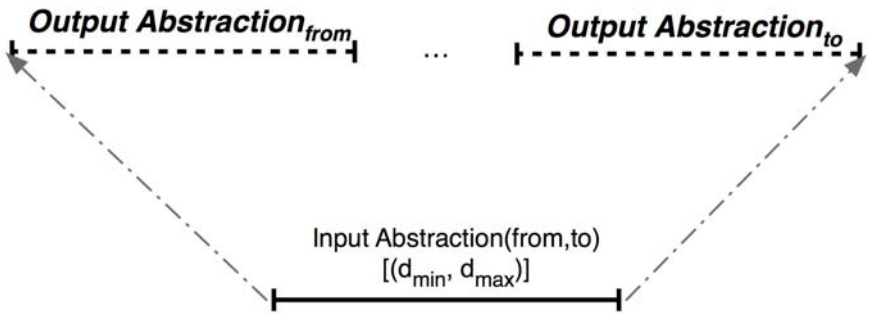


Figure 16: Temporal slice abstraction definition. Abstractions that use the temporal slice mechanism specify a previously defined abstraction, and an ordinal range (from,to). Constraints may be specified on the minimum and maximum duration of the input abstraction (d_{min}, d_{max}). The duration constraints are in square brackets to signify that they are optional.

3.3 IMPLEMENTATION

PROTEMPA was implemented using the Java Software Development Kit version 1.4 (java.sun.com) on Apple Macintosh hardware running OS X 10.3 (Apple Computer, Inc.), and was written in approximately 26,000 physical source lines of code. PROTEMPA's modules (Figure 8) are Java classes, and the PROTEMPA API can be called from any Java program. The processing sequence uses the Drools (www.drools.org) rules engine for abstraction finding. PROTEMPA has been successfully deployed on both Apple hardware running Mac OS X and standard PC hardware running Windows XP (Microsoft Corp.).

The back ends (Figure 8) are also Java classes; their class names are specified in a configuration file and loaded dynamically into the Java virtual machine. The Data Source backend implements a connection to a MySQL relational database (www.mysql.com). Two sets of Knowledge Source backend and Algorithm Source backend modules have been implemented for use in the evaluations of PROTEMPA described in Chapters 4.0 and 5.0.

3.3.1 Java backends

The first implementation specifies parameter definitions and algorithms as Java objects using PROTEMPA's knowledge and algorithm object models (Figure 13 and Figure 14), and serializes those objects to a file. This implementation was tested in an application of PROTEMPA for use in identifying abstractions in patient care (Chapter 5.0).

3.3.2 Protégé backends

In the second implementation, the Knowledge Source backend specifies a connection to a knowledge base implemented in the Protégé ontology environment (protege.stanford.edu). The Algorithm Source backend encodes algorithms as functions written in the R statistical

programming language (www.r-project.org), and stores the algorithms' source code as text in another Protégé knowledge base. The two knowledge bases' ontologies specify PROTEMPA's knowledge and algorithm object models, respectively. Protégé's ontology editor automatically generates user interfaces (not shown) based on these ontologies for creating and editing definitions of abstractions, time-stamped data types, and algorithms. Algorithms are read from the knowledge base, and executed as shown in Figure 17 using software from the RoSuDa project (stats.math.uni-augsburg.de/software/) that supports execution of R source code from Java programs. This implementation was tested in an application of PROTEMPA in clinical research described in Chapter 4.0.

3.4 DISCUSSION

PROTEMPA is a novel temporal abstraction framework that applies a two-stage strategy to detecting clinically important temporal patterns in raw patient data. The first stage implements a method based on TCV (77) that supports mathematical pattern matching with arbitrary algorithms called temporal abstraction primitives to infer low-level abstractions from time series data. The second stage applies iterative temporal pattern matching using TCNs (59) to infer high-level abstractions from groups of previously identified abstractions. In a proof-of-concept application of PROTEMPA, a set of simple domain-independent temporal abstraction primitives were developed for finding mathematical patterns of interest in clinical laboratory data, and a prototype of a web-based clinical laboratory data analysis tool has been implemented that visualizes intervals reflecting these patterns.

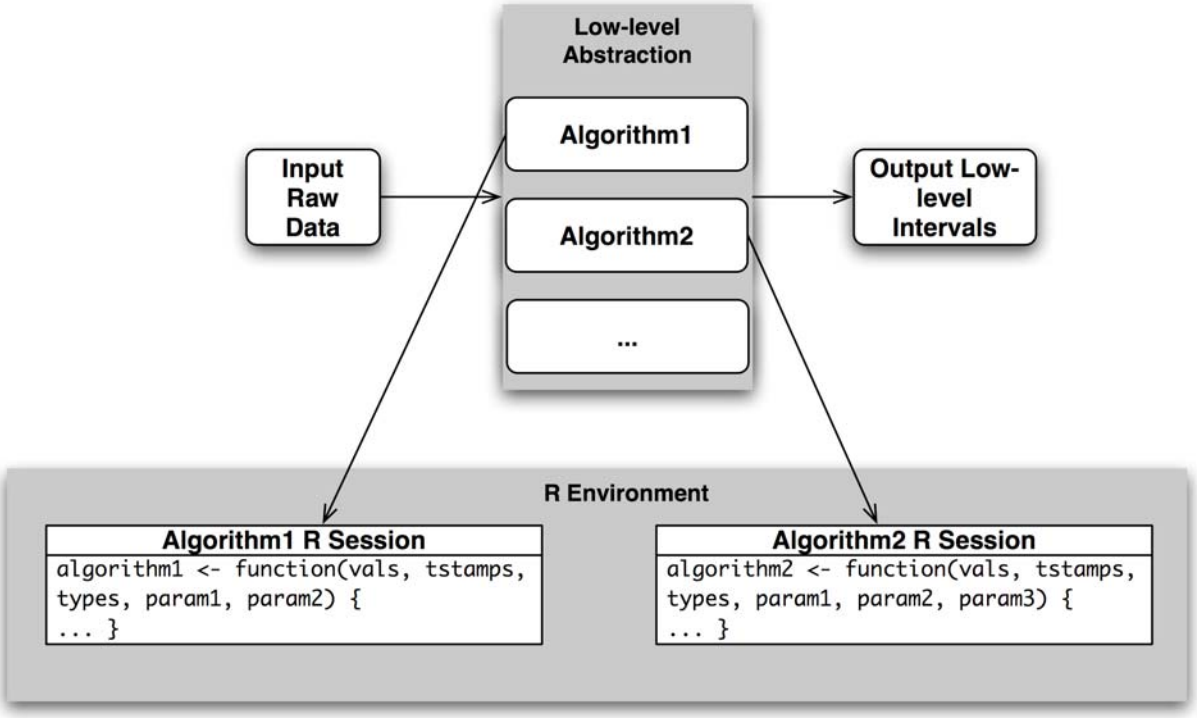


Figure 17: Illustration of data processing by the Algorithm Source, using the R Algorithm Source backend. After PROTEMPA's low-level abstraction mechanism selects a data subsequence for processing, it passes the subsequence into the Algorithm Source along with a set of algorithm-specific arguments. The Algorithm Source calls the appropriate algorithm and returns success if the data subsequence satisfies the algorithm's constraints. In the R Algorithm Source backend, each defined algorithm is executed in its own R session. *Vals* – the values of the data subsequence; *tstamps* – the timestamps of the data subsequence; *types* – the types of each data element of the subsequence; and *param1, param2, etc*. – algorithm-specific parameters.

The temporal abstraction primitive framework is designed to: 1) provide a clear separation between the mathematical pattern matching and temporal pattern matching mechanisms; 2) allow primitives to be evaluated and compared conveniently to determine the optimal algorithms for finding mathematical patterns in a particular domain; and 3) allow the use of arbitrary mathematical techniques for analyzing time series data. A wide variety of analysis techniques may be used to construct primitives, such as moving average (75), autoregressive (75), wavelet transform (131), and time series segmentation (82) among many others. This

implementation of primitives using a "plug-in" architecture is unique. Other temporal abstraction systems typically provide internal mechanisms for low-level abstraction with less flexible, pre-defined processing strategies.

Temporal abstraction primitives may be domain-independent or domain-specific. The *state* primitive is relatively domain-independent; classification based on value ranges is of interest across a wide variety of laboratory and other observations. The ability to set run-time parameters specific for primitive instances allows general-purpose domain-independent primitives to be tailored to specific needs. Analogous domain-independent primitives might evaluate event frequencies or trends in values associated with observations. Domain-dependent primitives that provide specific analyses for particular situations are also possible. For example, a primitive based on a particular compartmental drug elimination model might use that drug's dosing and blood concentration information to identify evidence for a specific pharmacogenetic phenotype. Furthermore, while the examples in this chapter involve clinical laboratory data values and drug administration data, it is possible to write primitives that operate across very different types of time-stamped data. For example, a primitive might detect patterns based on information extracted from free text clinical notes or reports using natural language processing.

Temporal abstraction primitives can be implemented in an arbitrary programming language and executed using an algorithm source backend that provides an appropriate runtime environment. Specifying algorithms in Java (Section 3.3.1) provides good performance, but may require a significant amount of programming if statistical processing is required. For applications that make extensive use of such processing, specifying algorithms in a statistical programming language such as R (Section 3.3.2) may provide significant reductions in the amount of time

required to implement new primitives. For example, R has a built-in linear regression model, making it trivial to implement a primitive for detecting trends.

PROTEMPA's separation of abstraction definitions, abstraction mechanisms, and algorithms for the low-level abstraction mechanism may significantly facilitate the maintainability of large sets of abstraction definitions, as compared with older rule-based expert system designs in which domain knowledge and data processing would be mixed together in the same rules (132). The separation of "temporal" domain knowledge (i.e. abstraction definitions) from mechanisms for computing temporal abstractions has been described previously for the RESUME system (Section 2.1.2) (133). PROTEMPA extends that work by separating temporal abstraction primitives from the low-level abstraction mechanism, thus making possible the independent creation and maintenance of these primitives as described above. Clinical domain experts who would define abstractions and raw data types do not need to know how the abstraction mechanisms and algorithms are implemented. Similarly, the low-level mechanisms and algorithms could be enhanced by software developers without having to modify any clinical domain knowledge. Separating domain knowledge from processing may also facilitate linking of PROTEMPA's knowledge base with existing standard clinical vocabularies that define raw data types, clinical states, and disease processes, as was implemented in ALMA (130). The Protégé environment (Section 3.3.2) facilitates both the entry of domain knowledge by clinicians and linking to external vocabularies, as was described by Musen in (134).

PROTEMPA is a hypothesis-testing system that scans time series data for pre-defined mathematical and temporal patterns of interest. This strategy is in contrast to a data mining tool that seeks to identify all meaningful patterns in a data set. A PROTEMPA hypothesis is a temporal abstraction that can be low-level or high-level, and domain-independent or -dependent;

the null hypothesis is the absence of the abstraction in the data set. PROTEMPA also supports the notion of simple hypotheses that aggregate to support higher-level hypotheses. This concept of hypothesis differs from that of TrenDx (14) in that a TrenDx hypothesis is a single result that is inferred to be the most likely explanation for a complex data set. Temporal abstraction systems such as RESUME (56, 65) and ALMA (130) do not have an explicit concept of hypotheses as such; their usual goal is to summarize a collection of data containing general-purpose, domain-independent abstractions to yield an accurate high level description of a patient's state over time.

Temporal abstraction has a worst-case algorithmic complexity that is exponential in the number of temporal relationships specified by a typical high-level abstraction (56). However, even systems that calculate all possible abstractions for summarization purposes usually need to specify only a limited set of temporal relationships, so average-case complexity is significantly better (56). Limiting abstraction definitions to hypotheses of interest may enable performance gains over summarization systems. PROTEMPA's two-stage system can further enhance performance over traditional temporal abstraction frameworks by allowing the implementation of more efficient algorithms for identifying abstractions than is possible in systems that build temporal relationship hierarchies with raw data elements at the leaves. Even TrenDx (14), which optimizes pattern detection on high frequency data by fitting raw data to regression models, may not be as efficient as time series segmentation algorithms (82), which cannot be implemented in TrenDx but can be in PROTEMPA. Finally, temporal abstraction is parallelizable per patient and has been successfully implemented on clustered hardware (13).

PROTEMPA can function as a retrospective analysis tool (scanning population data for patients with patterns of interest) or as a prospective data monitor (identifying patterns as they develop). The user chooses the "mode" simply by choosing the data; PROTEMPA functions

identically in both prospective and retrospective applications. Though a prospective analysis tool based on PROTEMPA has not yet been implemented, it would likely be straightforward to assess the performance of PROTEMPA pattern detection algorithms against retrospective data before deploying them in a prospective setting.

The abstractions that PROTEMPA detects, like those targeted by RESUME (56) and ALMA (130), do not yield a complete diagnosis, but are rather observations of clinically meaningful data patterns that are associated with the state of a patient or the nature of a clinical process (1). Identification of these abstractions may be useful in support of patient management tasks like therapy planning, determining prognosis, or monitoring response to therapy, and may be especially helpful for physician decision-support in the management of chronic disease where the primary diagnosis is already established. Recognition of intermediate abstractions is also likely to be useful in quality assurance or process control analyses, where particular patterns of events or values may indicate a process problem, a medical error or a "near-miss" (135, 136) (see Figure 7). In these settings, the aggregation of data through intermediate abstractions may identify features of the clinical course that are relatively obscure in the raw data, and may reduce "information overload" and focus decision-makers' attention on key issues.

Temporal abstraction systems lack support for attaching a probability to found intervals that the clinical state or process described by the interval actually exists (Section 2.1.7), and PROTEMPA is no different in that regard. This limitation is a by-product of temporal abstraction systems being of a rule-based expert system design. Using Bayes' theorem to calculate the probability of clinical states or processes occurring would theoretically have this capability, such as in the modifiable temporal belief network (MTBN) approach described in Section 2.1.7. However, representing states and processes as temporal and mathematical patterns in rules

appears to be a natural way of encoding these kinds of clinical knowledge, and assigning accurate prior or conditional probabilities of occurrence to many such interpretations would be difficult.

There are several ways in which intermediate abstractions could contribute to clinical decision support systems. Processing current clinical data to identify abstractions prospectively could support robust clinical alerting that is capable of recognizing developing temporal patterns within and across clinical event types. Such a system could support traditional message alerts delivered by pager, email or clinical information system user interface elements. Alternatively, intermediate abstractions could be used in combination with "presentation hints" or a display specification language (e.g., SmartDisplay (23)) to create dynamic clinical user interfaces that optimize themselves on-the-fly to most effectively display patterns in retrieved data. This potential is tested in the clinical user interface study described in Chapter 5.0. The abstraction hierarchies that PROTEMPA generates also support the development of interfaces that allow data to be visualized at multiple levels of abstraction (22), and PROTEMPA's data model supports the construction of a temporal reasoning interface for querying the features of discovered abstractions (57).

Although temporal abstraction systems have been shown to enhance medical care in several specific domains (16, 20, 65, 137-139), the possible benefits of a general-purpose temporal abstraction system in clinical event monitoring, clinical data display, medical quality assurance and outcomes research remain speculative. Additional studies are necessary to determine clinical situations that are amenable to this kind of analysis and optimize temporal abstraction primitives for best performance in those situations. PROTEMPA is designed to support these types of studies through a framework that accepts clinical data in a relatively

generic form, allows flexible design and iterative optimization of temporal abstraction primitives, and provides a principled mechanism for aggregating intervals into high-level abstractions.

4.0 IDENTIFYING CLINICAL PROCESSES IN RETROSPECTIVE DATA FOR PATIENT SELECTION

This chapter presents an implementation of PROTEMPA (introduced in Chapter 3.0) for retrospective data retrieval, and an evaluation study in which PROTEMPA was used to identify patients with HELLP syndrome and categorize them by disease severity and progression based on the temporal characteristics of multiple laboratory test profiles.

4.1 INTRODUCTION

Health care institutions store large volumes of clinical data that are useful for understanding processes associated with disease, therapeutic response and patient care (6). These processes may be reflected as time sequences of laboratory test results, medical observations, physiologic signals and other data that are mathematically and temporally related (1, 20, 25, 140, 141). Common clinical data retrieval systems do not provide tools for characterizing such data sequences or retrieving groups of patients who show them (5-10). Rather, they query for patients primarily by diagnosis, procedure, or billing codes; simple text matching; or limited mathematical comparisons of individual data values. Patient characteristics that are not explicitly coded, or are represented as mathematical patterns or temporal relationships within data sequences, are not easily accessible to these systems and thus are difficult to include in clinical, quality assurance or outcomes research studies.

The need for an improved ability to represent temporal knowledge and detect temporal relationships in clinical data sets has been recognized in previous data retrieval systems, but limited support exists in those systems for mathematical data processing. For example, DXtractor (4) is a tool for identifying patient populations in clinical databases. It supports specifying temporal relationships between time-stamped data values and mathematical constraints on single values, but it is not designed to detect mathematical patterns within data sequences. Temporal extensions to Structured Query Language (SQL) have been proposed (142) that support storage and retrieval of representations of clinical states and processes that occur over periods of time, but those systems require modification of existing database schemas. Furthermore, like DXtractor they do not detect mathematical patterns within data sequences. The Arden Syntax (143, 144) can represent decision rules for patient monitoring (3) that specify temporal relationships between individual data values. Detecting some mathematical patterns within data sequences is technically possible using the syntax and a limited set of mathematical functions defined by the Arden specification, but pattern detection algorithms must be specified separately within each rule. Thus algorithms cannot be shared between multiple rules or used as building blocks for complex rules.

Temporal abstraction, described in Section 2.1.2, employs an alternative approach for representing and identifying time relationships (12), and has been integrated previously with clinical databases. For example, Chronus II (145) is a recently-described temporal database architecture and query system that employs temporal abstraction (13) to retrieve patient populations from a retrospective data set. Like the temporal abstraction systems described in Section 2.1.2, the range of abstractions that can be created to describe patient features is limited to what can be constructed from pre-defined temporal abstraction primitives. Thus there remains

a need for software that can flexibly specify and identify a broad range of meaningful mathematical and temporal relationships within existing large clinical data sets stored as time-stamped records in standard relational databases.

To address this need, PROTEMPA (Process-oriented Temporal Analysis, see Chapter 3.0) has been applied to specifying temporal and mathematical relationships between data elements in standard time-stamped databases, and retrieving populations of patients who show those relationships. To test the hypothesis that temporal abstraction can be used to identify patients with specific disease processes and stratify those patients into disease severity and progression categories, this study evaluated PROTEMPA for its ability to 1) identify patients with HELLP syndrome in a large clinical data repository and 2) stratify those patients based on the characteristics of intervals in their clinical laboratory test results.

4.2 DESIGN OBJECTIVES

The goal of applying PROTEMPA to retrospective data analysis was to create tools that identify sequences of clinical events and observations based on mathematical constraints and temporal relationships, and retrieve patient populations containing those sequences from clinical data repositories. These tools should support a variety of tasks, including clinical research, and should not constrain future research questions. They should therefore provide substantial flexibility in specifying the mathematical and temporal relationships that are used to define data sequences. To simplify the maintenance of knowledge about data sequences and to allow reuse of definitions across multiple settings, the system should support creation and maintenance of an extensible library of temporal abstraction primitives (algorithms). These primitives may define

general mathematical patterns applicable to data sequences and should be configurable with parameters constraining them to sequences with particular characteristics. Because disease and clinical care processes may be reflected by multiple data sequences, each with characteristic features (e.g., time courses of change in several laboratory tests following a clinical procedure), the system must also support the specification of temporal relationships between groups of data sequences and abstraction of those groups into higher-level time intervals. To be broadly useful, the design must support interoperability with existing data stores and integration into standard networked computing environments.

4.3 HELLP SYNDROME

HELLP (Hemolytic anemia, Elevated Liver enzymes, and Low Platelets) is a severe form of pre-eclampsia that appears during the latter part of the third trimester or after childbirth (39). There is no standard diagnosis code (ICD-9) for HELLP. Diagnosis and management are based on monitoring clinical symptoms and three clinical laboratory tests: platelet count (PLT), lactate dehydrogenase (LDH), and aspartate aminotransferase (AST). HELLP syndrome has been defined as pre-eclampsia with PLT < 100,000/μL, LDH > 600 U/L, and AST > 70 U/L (146, 147), and rising PLT indicates recovery (147). The PLT nadir can be used to classify a HELLP patient by disease severity: class 1 HELLP, PLT < 50,000/μL; class 2 HELLP, PLT >= 50,000/μL and < 100,000/μL (148). There is clinical interest in the characteristics and therapeutic circumstances of HELLP patients who show a partial recovery of PLT followed by a second suppression. Formerly, identifying these patients required extracting pregnant patients by diagnosis code and inspecting their laboratory results by hand.

4.4 METHODS

PROTEMPA's design and implementation were described in Sections 3.2 and 3.3, respectively. For this application of PROTEMPA, the Data Source backend implements a connection to a MySQL relational database (www.mysql.com), and the Protégé Knowledge Source and Algorithm Source backends were used as described in Section 3.3.2.

To determine whether PROTEMPA could be used to identify and categorize patients with laboratory signs of HELLP, a set of cases was retrieved from the University of Virginia's Clinical Data Repository (CDR) (10) occurring between 2000 and 2005 with ICD-9 codes indicating pregnancy and at least one LDH > 300 U/L. All available laboratory test results and diagnosis codes for each case were exported from the CDR and imported into the test implementation's data store. Laboratory data were originally obtained as part of routine clinical care using standard analysis methods. Institutional Review Board approval was obtained.

Five temporal abstractions were defined that distinguish between the two disease severity categories (*HELLP 1* and *HELLP 2*, as defined above) and four PLT response categories: 1) those that recovered after PLT suppression (*First recovering)*, 2) those that partially recovered, recurred and then recovered (*Recurring with recovery*), 3) those that partially recovered, recurred and then did not recover (*Recurring*), and 4) those that showed no evidence of recovery (all others). These five abstractions incorporate general temporal abstraction primitives for detecting states, trends, and the minimum value in a data sequence, with arguments specific for PLT, LDH, and AST.

The *HELLP 1* and *HELLP 2* abstractions use the temporal pattern mechanism (Section 3.2.2), and specify intervals of elevated AST, elevated LDH, and low PLT occurring within 7 days of each other with a minimum PLT value of less than 50,000/μL or between 50,000/μL and

100,000/μL, respectively. The *First recovering* abstraction uses the temporal slice (Section 3.2.2) and temporal pattern mechanisms. It specifies the first interval of platelet values, beginning after the start of a class 1 or 2 HELLP interval, that shows an increase of more than 9,000/μL per day and is at least 12 hours in duration. The *Recurring* abstraction uses the temporal pattern mechanism and specifies an interval of *First recovering* followed by an interval of platelet values with a decrease of more than 9,000/μL per day to an endpoint of less than 100,000/μL. The *Recurring with recovery* abstraction uses the temporal pattern mechanism and specifies an interval of *Recurring* followed by an interval of platelet values with an increase of more than 9,000/μL per day to an endpoint of at least 100,000/μL. An example of a patient case showing these abstractions is illustrated in Figure 18, and the definition of the *Recurring* abstraction is illustrated in Figure 19.

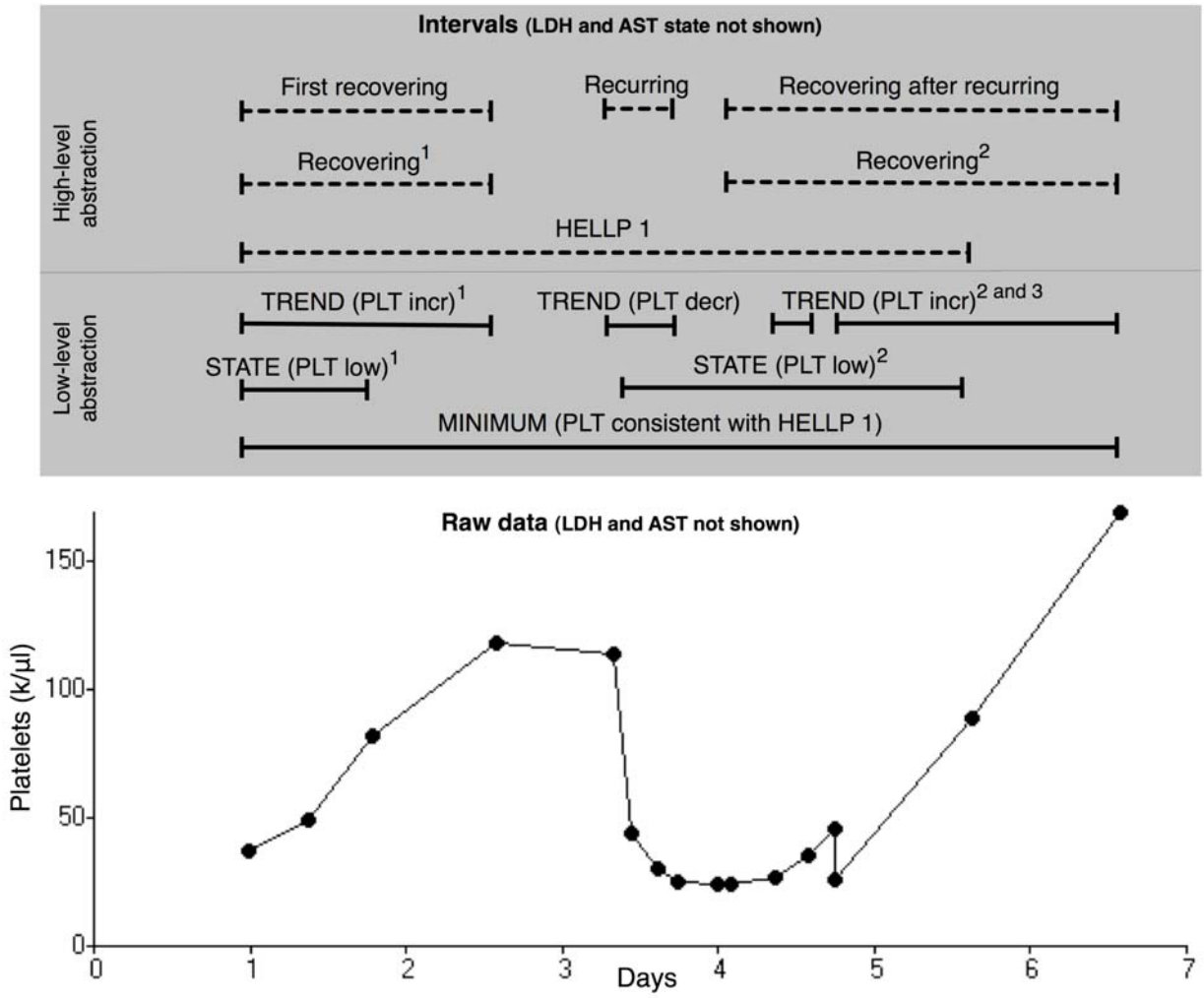


Figure 18: Example time series of platelet (PLT) counts in class 1 HELLP syndrome (39, 148) with the intervals that PROTEMPA identified. Intervals found by the low-level abstraction mechanism (solid lines) were inferred from mathematical patterns in the raw time-stamped data shown at the bottom of the figure, and are labeled as in Figure 11. Intervals found by the high-level mechanism (dashed lines) were identified as in Figure 19 and are labeled with the corresponding abstraction's name. Superscripts refer to the ordinal position of the intervals accessible through the temporal slice mechanism (see Design). AST and LDH test results and associated intervals are omitted for clarity.

A simple Java application invoked the PROTEMPA library with database connection information, Protégé knowledge base files defining the abstractions and algorithms, and a list of abstractions to find; and output found intervals for each case to a text file. The abstraction definitions were adjusted after a preliminary processing run to optimize patient categorization and accommodate typical variations in laboratory values. After PROTEMPA identified a set of cases and their abstractions, its output file was passed to a post-processing script, which categorized the cases according to the types and sequence of intervals found (Table 1). The HELLP cases found by PROTEMPA, and their categorizations, were compared for accuracy against manual review of the data set by the author. The HELLP cases found by PROTEMPA were also compared to those identified by a standard SQL query looking for cases with at least one elevated value of AST, one elevated value of LDH, and one suppressed PLT result. This query is intended to be representative of the way existing clinical data retrieval systems would be used to search for HELLP cases.

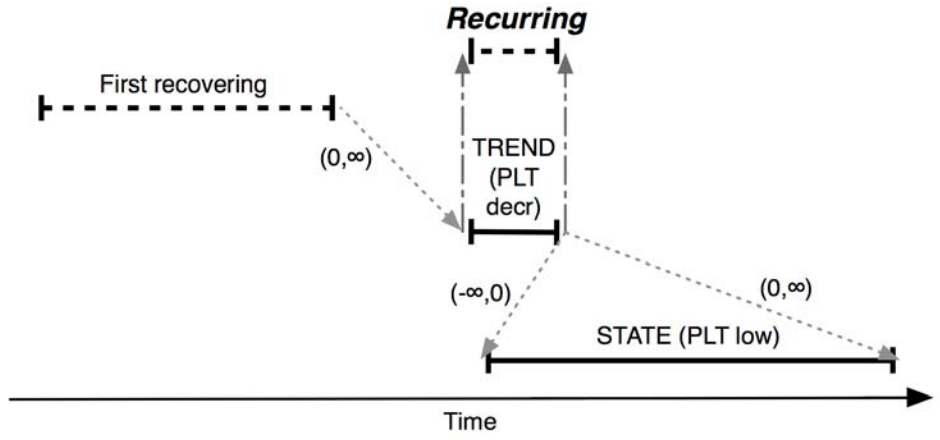


Figure 19: Composition of the *Recurring* abstraction (see Methods), defined as an interval of decreasing platelet values (*PLT decr*) occurring after the first interval of recovering platelets (*First recovering*) and overlapping an interval of platelet values less than 100,000/μL (*PLT low*). Intervals are illustrated as in Figure 18. Gray dotted arrows denote temporal relationships defined between endpoints of intervals and are labeled with minimum and maximum time constraints (see Figure 15 for details on the temporal relationship notation). When *First recovering*, *TREND(PLT decr)*, and *STATE(PLT low)* intervals are found with these relationships, a *Recurring* interval is created with the same endpoints as the contributing *TREND(PLT decr)* interval (gray dashed arrows).

Table 1: HELLP syndrome cases categorized by laboratory result profile.

Severity	Category	Number (percent of total)
HELLP 2[1]	Recurring[2]	5 (6.2%)
	Not recurring[3]	33 (40.7%)
	Total	38 (46.9%)
HELLP 1[1]	Recurring with recovery[4]	7 (8.7%)
	Recurring without recovery[5]	4 (4.9%)
	Not recurring with recovery[3]	31 (38.3%)
	Not recurring without recovery[6]	1 (1.2%)
	Total	43 (53.1%)
All HELLP	Total	81 (100%)

[1]Patients were categorized as HELLP 1 or HELLP 2 based on co-occurring intervals of elevated LDH and AST, and suppressed platelet count (see Evaluation). All HELLP 2 patients recovered based on normalization of platelet counts.
[2]Final platelet interval is *Recurring* or *Recurring with recovery*
[3]Final platelet interval is *First recovering*
[4]Final platelet interval is *Recurring with recovery*
[5]Final platelet interval is *Recurring*
[6]All patients not classified as above

4.5 RESULTS

There were 761 eligible cases (pregnancy and elevated LDH) of which 190 included an ICD-9 code for severe pre-eclampsia. PROTEMPA took five minutes to process the 761-case data set, and identified 81 cases as likely HELLP. All of the cases that PROTEMPA found were confirmed by manual inspection as consistent with HELLP. The post-processing step correctly classified the 81 potential HELLP cases into severity and platelet response pattern categories (see Table 1), as compared with manual inspection of each case's PLT data sequence. Overall, PROTEMPA identified the "recurring" pattern (partial platelet recovery followed by a subsequent platelet suppression) in about 20% of HELLP patients (6.2% of HELLP 2 and a total of 13.6% of HELLP 1 patients, Table 1).

The standard SQL query identified 87 of these 190 cases as potential HELLP diagnoses based on the presence of at least one low PLT, high AST, and high LDH consistent with the HELLP definition. In all six SQL-identified cases that PROTEMPA did not identify, laboratory results meeting the required thresholds were more than 7 days apart and thus were likely to be unrelated. It was not possible to classify response patterns using SQL queries because SQL does not offer the capability to distinguish time course details.

4.6 DISCUSSION

PROTEMPA is a data processing strategy and software library that allows specification and identification of mathematical and temporal relationships in clinical data, and recognition of patients in retrospective clinical data repositories based on these relationships. This study tested PROTEMPA in a proof-of-concept implementation that accurately identified patients with

laboratory signs of HELLP syndrome, a disease with no ICD-9 code, and accurately assigned severity and disease progression categories based on the temporal characteristics of laboratory test profiles (platelet count, AST, and LDH; see Figure 18 and Figure 19). The ability to automate identification of patients and processes of interest based on the temporal characteristics of clinical data may substantially decrease the effort required to retrieve and classify patients for a wide range of clinical studies, outcomes research, and quality assurance evaluations. This is particularly pertinent in cases involving patient features that do not have standard codes, in which coding may not be complete or accurate, or which require discrimination of clinical severity, disease progression or response to therapy. Though some previous data retrieval systems allowed queries with temporal features (4, 5, 142), PROTEMPA is unique in supporting the identification of patient populations based on multiple data sequences with defined characteristics and temporal relationships.

The comparison of patients identified by PROTEMPA versus those identified by a standard data retrieval system query illustrates how existing methods of identifying patients by setting constraints on the values of multiple laboratory tests do not take the temporal distance between results into account. In this data set, the SQL query described above resulted in six false positive HELLP cases, but there might be more in a different patient population or clinical problem. These illustrate the type of false positives that can occur if temporal patterns are not considered. Another significant advantage of PROTEMPA is its ability to identify patterns in laboratory result profiles that are indicative of disease severity and progression, a capability that SQL does not support.

The temporal abstraction primitives that PROTEMPA uses in its low-level mechanism (Figure 11) are particularly suited to identifying mathematical patterns for retrospective data

retrieval. Primitives may be fully specified ("hard coded") algorithms, or they may be designed to receive parameters that define the features of the data sequences they match. The latter supports the identification of data sequences that have similar general features (e.g., *state*, *trend*, *peak*, *trough*, *frequency*) but may vary in specific characteristics such as cutoff values, slopes, total number of data elements and duration in different clinical settings. PROTEMPA primitives are designed to be flexible; in addition to identifying relatively short mathematical patterns, they could also target longer data sequences using a variety of analysis techniques. Algorithms may be of arbitrary complexity and may be implemented in any programming language, as long as an appropriate algorithm specification and runtime environment are provided by the Algorithm Source backend (Figure 8). The R statistical processing environment (Figure 17) proved flexible and convenient for the purposes of this study. Other interpreted or compiled programming environments designed for statistical, scientific or modeling applications could also be implemented as Algorithm Source back ends as appropriate for particular tasks.

PROTEMPA's performance characteristics for retrospective data analysis have not yet been fully evaluated. As described in Chapter 3.0, its data processing mechanism (Figure 12) has exponential worst-case complexity, but is expected to have significantly better average case complexity. Previous studies have found that patient summarization and monitoring tasks usually require searching for a limited set of abstractions (56), and the abstractions developed for identifying HELLP cases in this study indicate that clinical research questions may be similar in that respect. For large data sets or high search volumes, PROTEMPA's algorithms are parallelizable by patient (13) and could be implemented in a straightforward manner on clustered hardware. Performance might be further enhanced by parsing the abstraction definitions for

value constraints that might be used to limit the amount of data extracted from the Data Source backend.

4.7 CONCLUSIONS

PROTEMPA is a temporal abstraction system that can be used as a flexible, extensible retrospective data retrieval system and can accomplish tasks beyond the capabilities of other software for retrospective query of clinical systems. In this preliminary study, it correctly identified and categorized patients with a complex disease based on temporal relationships between multiple laboratory results. Temporal abstraction has not been commonly applied to clinical data retrieval, but it may provide significant advantages when used to augment standard data retrieval methods in clinical research, outcomes studies and quality assurance. It is likely to be particularly useful in situations where the patient characteristics of interest are not easily coded or are expressed as changes in data over time, temporal relationships between multiple data elements, or frequencies of events.

5.0 CLINICAL DECISION-MAKING USING A PROCESS-ORIENTED DATA DISPLAY

This chapter introduces a clinical data display that implements the visualization component of the process-oriented approach for improving the detection and visualization of disease and patient care processes (see Chapter 1.0). After describing the display's design and implementation, this chapter describes an evaluation of the effects of a process-oriented display of clinical laboratory data on data acquisition, interpretation, and decision-making by physicians, as compared with a traditional numerical laboratory data display.

5.1 INTRODUCTION

Clinical laboratory results support many clinical decision-making activities, including diagnosis, monitoring disease progress, and measuring response to therapy. Existing production medical record systems typically present laboratory results similarly to the static presentations found in the paper chart, and thus do not take advantage of computer systems' capacity for data processing and dynamic visualization. Such presentations of laboratory data are probably suboptimal and may contribute to medical errors (see Section 2.3).

Previously reported interview data (described in (93), and discussed in Section 2.2.2) suggests that physicians reviewing real cases primarily use other physicians' accounts of a patient in physician-written documents for the purpose of obtaining an overview, and use

"primary" data sources such as laboratory displays for evidence seeking. These results suggest a need for evaluating data displays in a full medical record context to ensure that each component of the display is optimized for its real use in clinical workflows.

Displays with concept-oriented organization and/or graphical display of clinical data have been found in limited studies to improve interpretation and decision-making by physicians (see Sections 2.2.1 and 2.3). Clinical data displays that graphically visualize intervals found by temporal abstraction have been prototyped (21-23) (see Section 2.2.1), and limited evaluations in highly controlled settings suggest that they improve the speed of data acquisition and interpretation by physicians (see Section 2.2.4 and 2.3). A medical record system, the Physician's Workstation (126), dynamically highlights clinically relevant trends and states in laboratory results (see Section 2.3), but the effects of emphasizing these patterns on decision-making has never been evaluated. Concept-oriented organization, graphical display, and pattern visualization have the effect of highlighting relationships in the time-stamps and values of clinical data that are reflected by clinically relevant patient processes. There is a need to determine whether emphasizing data reflecting these processes improves clinical decision-making by physicians in context.

To evaluate whether the process-oriented approach improves data acquisition and/or clinical decision-making by physicians, a data display was developed for results review called TPOD (Temporal Process-oriented Display). TPOD dynamically analyzes clinical data and adapts its form, organization, and sequence to highlight clinically relevant patient processes. It graphically visualizes temporal and mathematical patterns within time sequences of patient data, organizes data related to these patterns in close proximity, and sequences data to prioritize the display of found patterns. TPOD is implemented in a web-based case presentation software

system that dynamically organizes the display of clinical laboratory data based on temporal and mathematical patterns found by PROTEMPA.

To evaluate TPOD's effects on the process and outcome of data acquisition, data interpretation, and clinical decision-making by physicians, a study was conducted in which resident physicians evaluated a set of real patient cases and wrote orders using case presentation software that includes a variety of clinical data, and presents laboratory data with either a TPOD-based display or a tabular numerical display. Subjects were asked to "think-aloud" while reviewing cases, and their verbal utterances and on-screen actions were recorded. The case presentation software recorded subjects' orders, navigations from screen to screen, and time spent on each case.

Two analyses of these data were performed. Given the lack of previous studies evaluating clinical data displays in context, a preliminary analysis examined "think-aloud" transcripts collected from the cases assigned to the tabular display in order to ascertain how physicians use laboratory results during case review and decision-making. As physicians appear frequently to use others' accounts of a patient case for obtaining an overview (see above), this initial analysis hypothesized that components of the medical record that provide primary data may be more often used for obtaining evidence of known problems and identification of anomalous findings than they are for overview.

The second analysis compared the process and outcome of decision-making in cases with a TPOD-based laboratory display versus cases with a tabular display. Specifically, this analysis compared the quality of the orders that subjects wrote, the time required to complete the review and order set for each case, the time spent using the laboratory display, the frequency with which the laboratory data display was used, the cognitive steps that subjects took while reviewing the

cases, and subjects' opinions of the laboratory displays. Hypotheses of this analysis were that TPOD would improve the speed and accuracy of orders, decrease the ratio of information acquisition to information integration and evaluation steps, decrease the time required to evaluate a case, and decrease the total number of cognitive steps as compared with the tabular display of laboratory results.

5.2 TPOD: TEMPORAL PROCESS-ORIENTED DISPLAY

TPOD has a client-server architecture (see Figure 20). The client is a web browser. On the server side, a *patient database* stores time-stamped patient data. A *knowledge base* specifies how the display dynamically alters its form, organization, and sequence in response to PROTEMPA abstractions. A *process detector* implements PROTEMPA's Abstraction Finder (see Section 3.2.2). The *display server* uses the intervals identified by PROTEMPA and information about those intervals in the knowledge base to format a patient's data for display.

The knowledge base specifies TPOD's form, organization, and sequence as clusters of data that are displayed if a specified group of temporal and mathematical patterns is present (see Figure 21). These *data clusters* are specified as: a PROTEMPA *abstraction* that must be present in order for the cluster to be displayed, a collection of *supporting data types* that may be optionally present and may help clinicians understand the significance of the cluster, and a *priority rank* that determines how prominently the display should present the cluster. Constraints may be specified on the minimum and maximum length of the abstraction. Data clusters may contain overlapping primary or supporting data, in which case the data would be visualized in all clusters in which they are specified. When an abstraction specified by a data cluster is present,

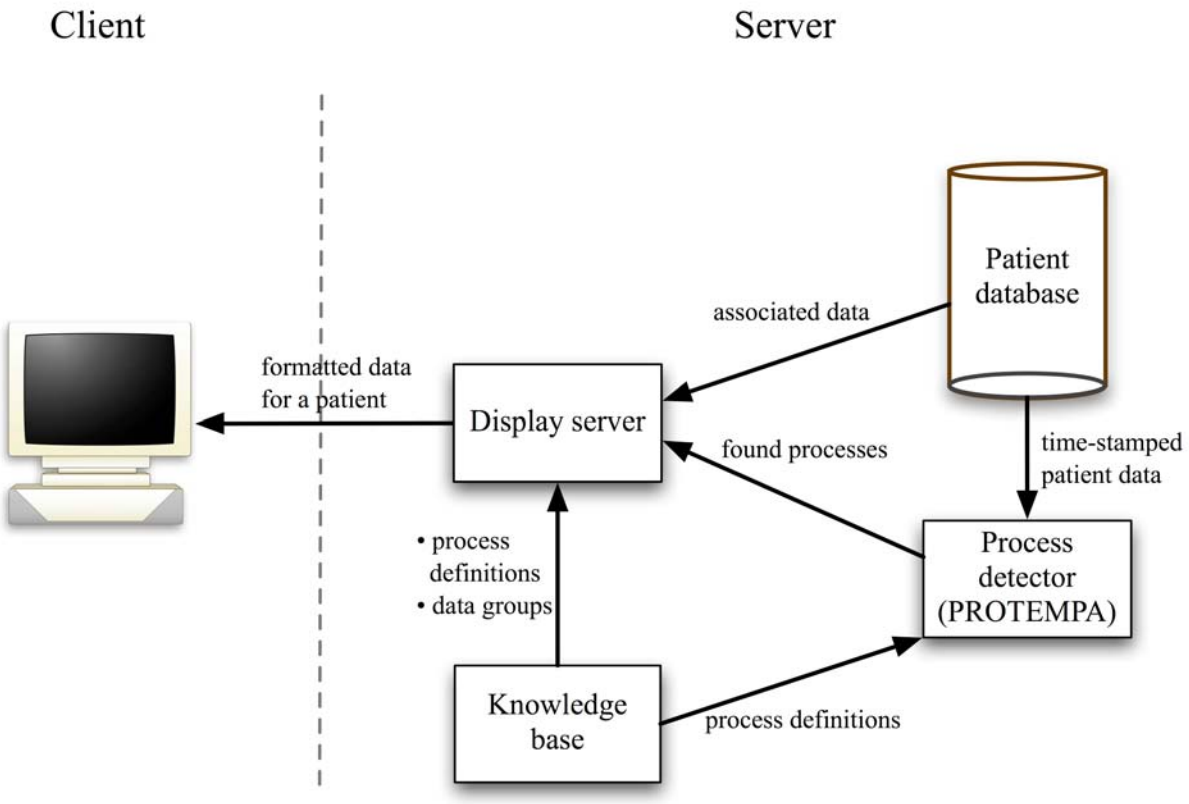


Figure 20: Architecture of TPOD. TPOD has a client-server architecture. The client is a web browser. The server has four components: a *knowledge base* for specifying display organization, form, and sequence; a *patient database*; a *process detector* for identifying abstractions in patient data; and a *display server* that presents intervals and data in graphical form.

the cluster is displayed as a graphical visualization of the patterns specified by the abstraction, graphical presentations of the data from which those patterns were derived (*primary data*), and additional data that may help clinicians understand the significance of the patterns (*supporting data*). Cluster displays are designed to aggregate data related to a clinical process or problem, graphically reveal temporal relationships between those data, and sequence data so as to highlight a patient's current clinical problems.

When the client requests data on a patient, the data server retrieves the set of defined data clusters from the knowledge base, and the process detector finds all instances of the abstractions

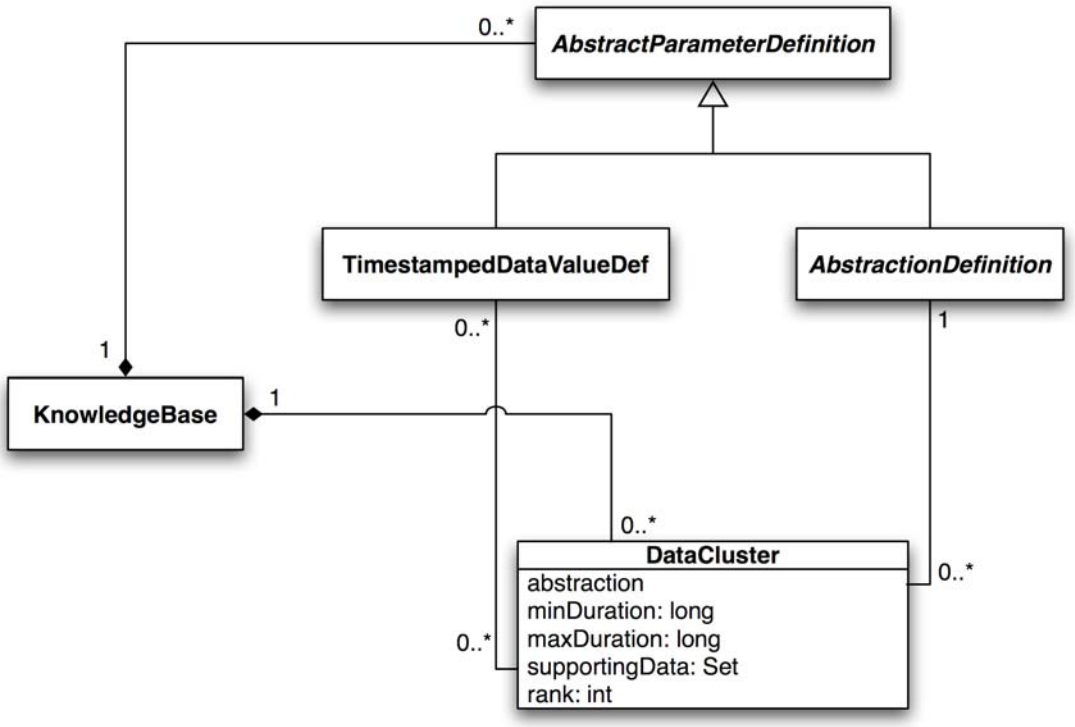


Figure 21: Class diagram of TPOD's knowledge model. This model extends PROTEMPA's knowledge model (see Figure 14) by including a *DataCluster* class, which represents a cluster of data that is displayed only if a specified abstraction is present.

defined in the data clusters, accessing the patient database as needed. For the data clusters whose abstraction has been identified, supporting data are retrieved from the patient database, graphs are created to visualize the data clusters, and a web page is returned to the client that displays data clusters in rank order.

TPOD is implemented as a collection of Java Servlets and JavaServer Pages (java.sun.com) generating HTML. Visualizations are created as image files using a customized version of the open-source JFreeChart charting software (www.jfree.org/jfreechart). TPOD's Process Detector implements PROTEMPA as described in Section 3.3. PROTEMPA's Data Source is used by TPOD for connecting to a patient database. The Knowledge Source and

Algorithm Source backends are implemented as described in Section 3.3.1. A Java applet (not shown) provides a user interface for specifying data clusters.

An example TPOD display is shown in Figure 22. The patient has six data clusters, displayed in rank order. The mostly highly ranked cluster, *Cholestatic Liver Disease*, is shown at the top and is fully displayed by default. Remaining clusters are initially hidden, but can be revealed by clicking on either the cluster's name or on the disclosure triangle to the left of the cluster's name.

The *Cholestatic Liver Disease* cluster is displayed if PROTEMPA identifies an abstraction specified as an interval of contemporaneous elevated values of ALT, Alkaline Phosphatase (AlkPhos), and GGT (the definition of this cluster is discussed below and is shown in Table 2). As shown Figure 22, the temporal extent of the interval is visualized as a horizontal bar (see Section 2.2.4 and Figure 5), and the data contributing to the interval are visualized as trend plots. Reference ranges and abnormal test results are highlighted. The *Cholestatic Liver Disease* cluster specifies four associated tests (see Table 2) that may be useful for interpreting the cholestatic liver disease process, and these are visualized as vertically compressed graphs (see Figure 22) that can be expanded to full-size by clicking on the magnifying glass to the right of each graph. The vertically compressed graphs have additional highlighting behind abnormal values so that clinicians can determine if the associated tests are abnormal at a glance, and multiple closely spaced abnormal results are given further emphasis with darker highlighting.

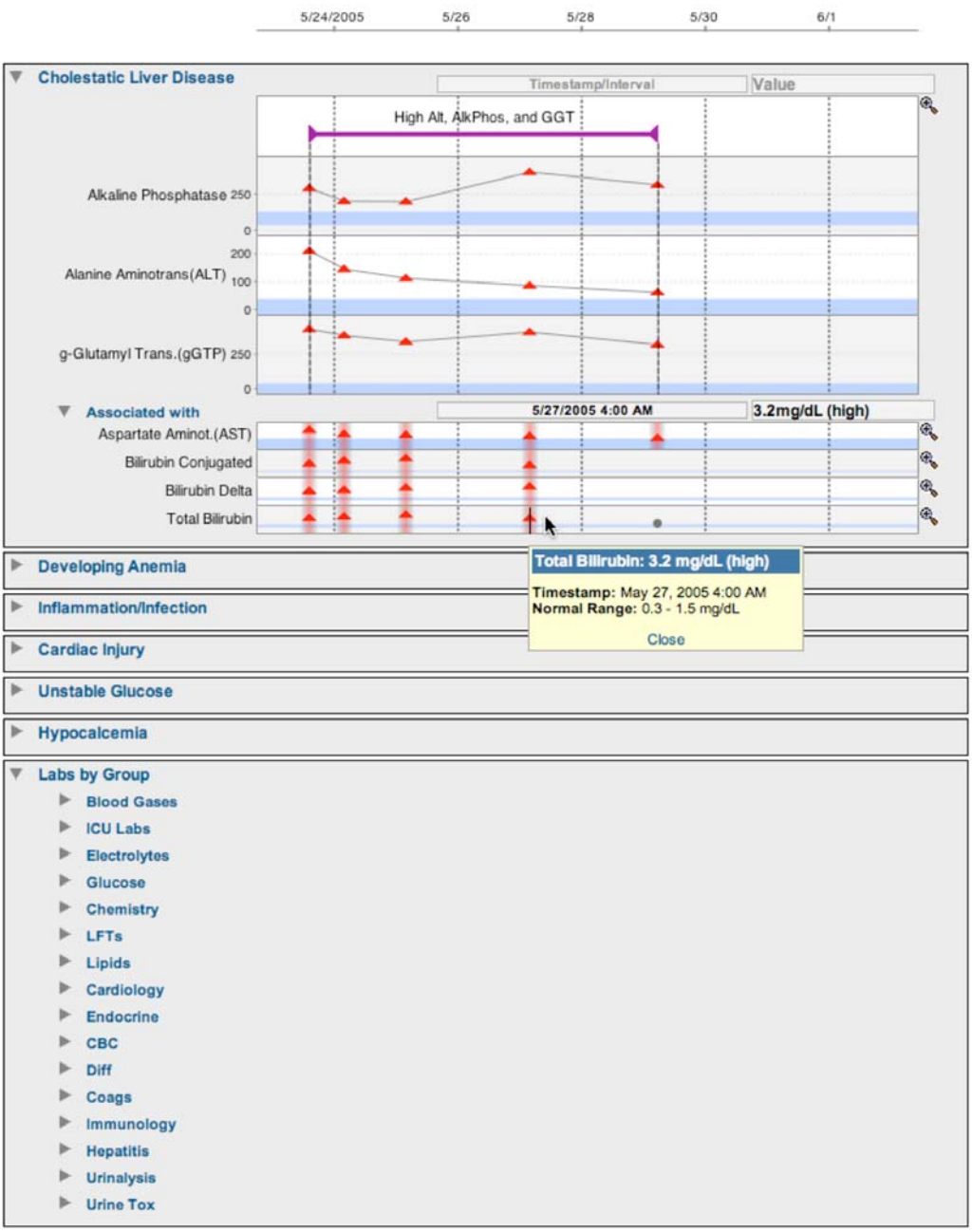


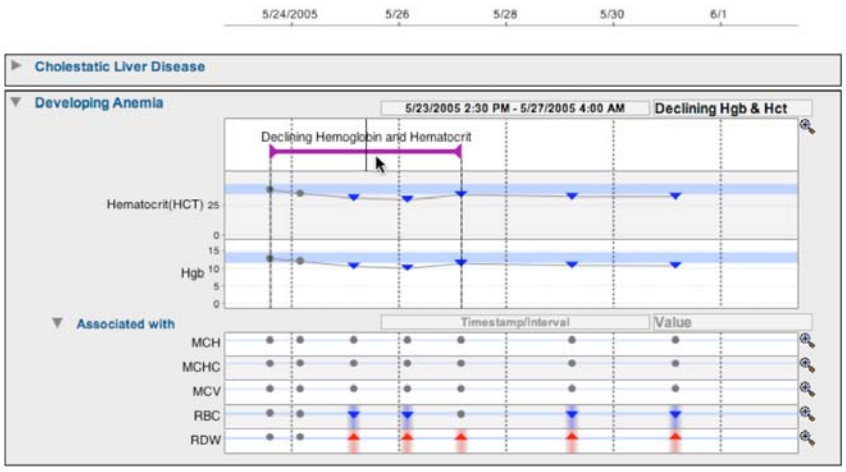
Figure 22: Screenshot of TPOD showing a *Cholestatic Liver Disease* data cluster (defined in Table 2), which is displayed because PROTEMPA found an interval of *High Alk, AlkPhos, and GGT* (purple bar). Data contributing to this interval are shown in full-size graphs, and associated tests are shown in compressed graphs. Each test's normal range is indicated in blue. Abnormal high tests are indicated with upward-pointing red triangles, and abnormal low tests are indicated with downward-pointing blue triangles (not shown). In the compressed graphs, abnormal results are further highlighted with gradient backgrounds that "coalesce" if sequential abnormal results are closely spaced to give those results greater emphasis. The mouse pointer is closest to a total bilirubin test result, and the result's timestamp (5/27/2005 4:00 AM) and value (3.2mg/dL) are displayed in boxes above the graph. Clicking the same bilirubin result causes a sticky-note widget to appear below and to the right of the result with detailed information.

While TPOD visualizes laboratory results graphically by default, access is also provided to numerical results. Moving the mouse pointer over a graph causes the timestamp and value of the laboratory result closest to the pointer to be displayed in the gray boxes over the graph (shown in Figure 22). Alternatively, by clicking on a test result, a "sticky-note" box pops up with detailed information about a result (Figure 22).

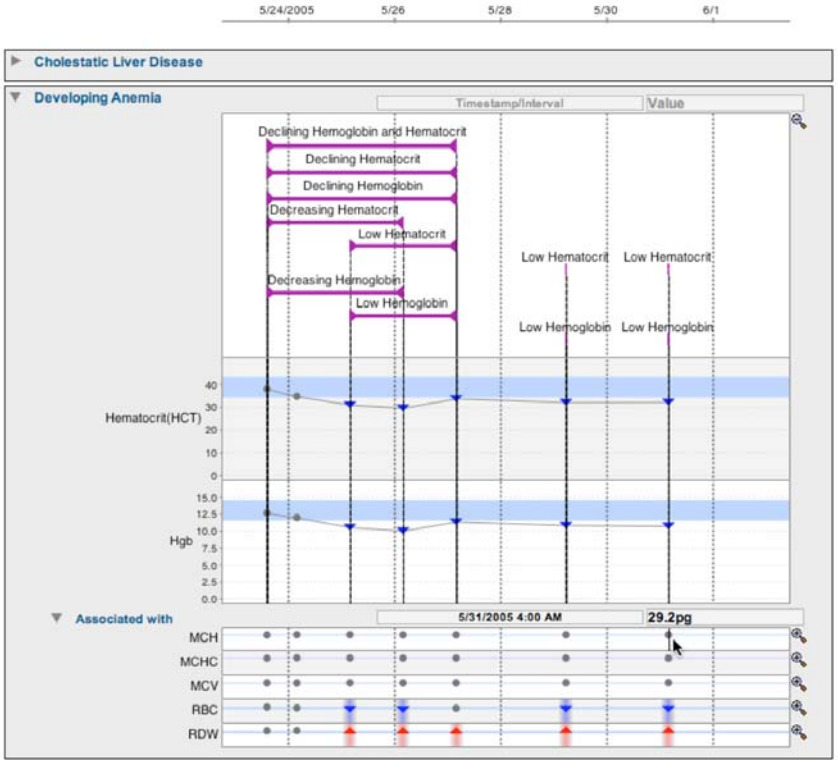
Data clusters only show the subset of a patient's data that is relevant to the abstractions that PROTEMPA has identified. Access to the rest of a patient's data is provided underneath the data clusters in a section of the display called *Labs by Group* (Figure 22). This section visualizes all of a patient's laboratory results as compressed graphs organized in static groups similarly to typical production laboratory displays.

If a cluster's abstraction is a high-level abstraction (see Section 3.2.2), TPOD can reveal the found interval's component intervals graphically to explain the basis for identifying the cluster. Figure 23A shows a cluster called *Developing Anemia* (defined in Table 2 and illustrated in Figure 28) that was displayed because PROTEMPA found an interval of *Declining Hemoglobin and Hematocrit* (defined in Table 2) using the temporal pattern mechanism (Section 3.2.2). By clicking on the magnifying glass to the right of the purple bar, TPOD shows all of the intervals from which *Declining Hemoglobin and Hematocrit* was derived (Figure 23B).

Thus, TPOD organizes clinical data according to temporal and mathematical patterns that have been found by PROTEMPA. It graphically visualizes the temporal extent of these patterns, the data contributing to these patterns, and additional data that may be useful for determining the clinical significance of these patterns. Data presented in these clusters are sequenced at the top of the display to give those data emphasis. TPOD also provides visual explanation for why a cluster has been displayed.



A



B

Figure 23: Screenshot of TPOD showing a *Developing Anemia* data cluster (defined in Table 2). A. Initial visualization of the cluster, showing an interval of *Declining Hemoglobin and Hematocrit* (purple bar). Clicking on the magnifying glass icon to the right of the interval reveals an expanded visualization. B. The expanded visualization, which is read from top to bottom. It shows that the interval of *Declining Hemoglobin and Hematocrit* was derived from intervals of *Declining Hematocrit* and *Declining Hemoglobin*; *Declining Hematocrit* was derived from intervals of *Decreasing Hematocrit* and *Low Hematocrit*; and *Declining Hemoglobin* was derived from intervals of *Decreasing Hemoglobin* and *Low Hemoglobin*. The two *Low Hematocrit* and two *Low Hemoglobin* intervals on the right-hand-side of the visualization did not contribute to the *Declining Hemoglobin and Hematocrit* interval.

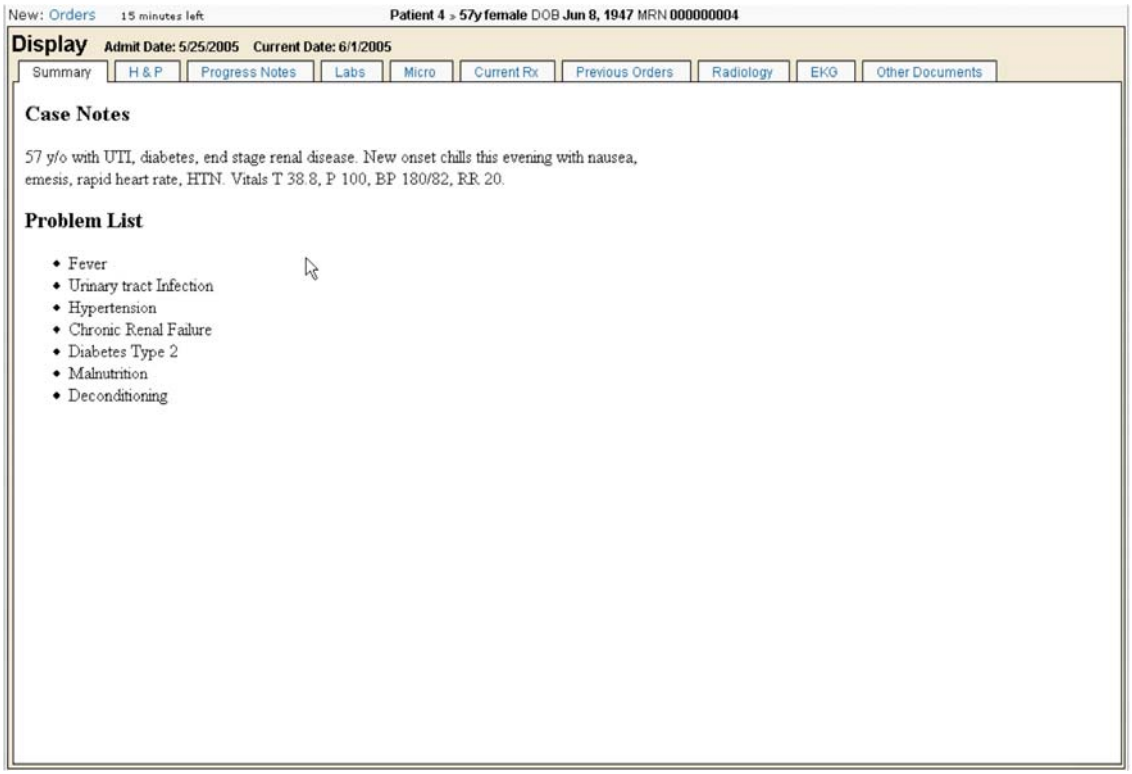


Figure 24: Screenshot of the case presentation software's Summary tab.

5.3 METHODS

5.3.1 Case construction

Cases were real, de-identified patient data extracted from the University of Pittsburgh Medical Center's data warehouse, MARS (Medical Archival System, www.mars-systems.com). Institutional Review Board approval was obtained. An honest broker obtained a frequency distribution of ICD-9 discharge codes for all hospital admissions of at least seven days duration. From that distribution, twelve ICD-9 codes were selected based upon two criteria: 1) a diagnosis for which laboratory testing is important for patient management; and 2) a code for which at least

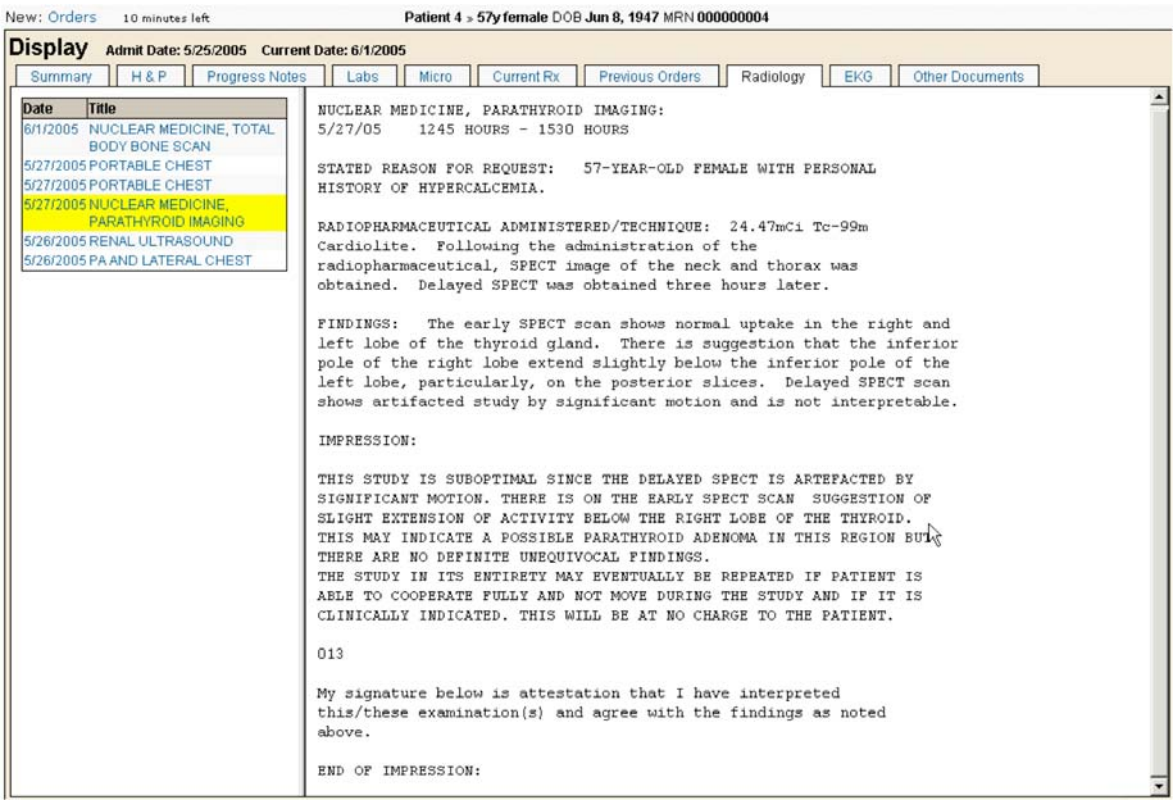


Figure 25: Screenshot of the case presentation software's Radiology tab. The H & P, Progress Notes, Micro (Microbiology), EKG (Electrocardiogram), and Other Documents tabs have a similar appearance.

80 cases were present. Eight cases were randomly chosen for each ICD-9 code, for a total of 96 candidate cases.

The candidate cases were screened for several criteria: 1) a complete admit note or a first progress note sufficiently detailed to be used as an admit note; 2) a discharge summary; 3) a reasonably complete set of progress notes, laboratory results, and consultation reports; 4) visit of sufficient complexity to test the two display styles. Complexity was calculated as the sum of the number of lines of text, number of orders and number of laboratory results for each case. Twelve cases met these criteria.

Four internal medicine attending physicians were each given two or three cases, with instructions to select a stopping point to make each case between six and eleven days in length

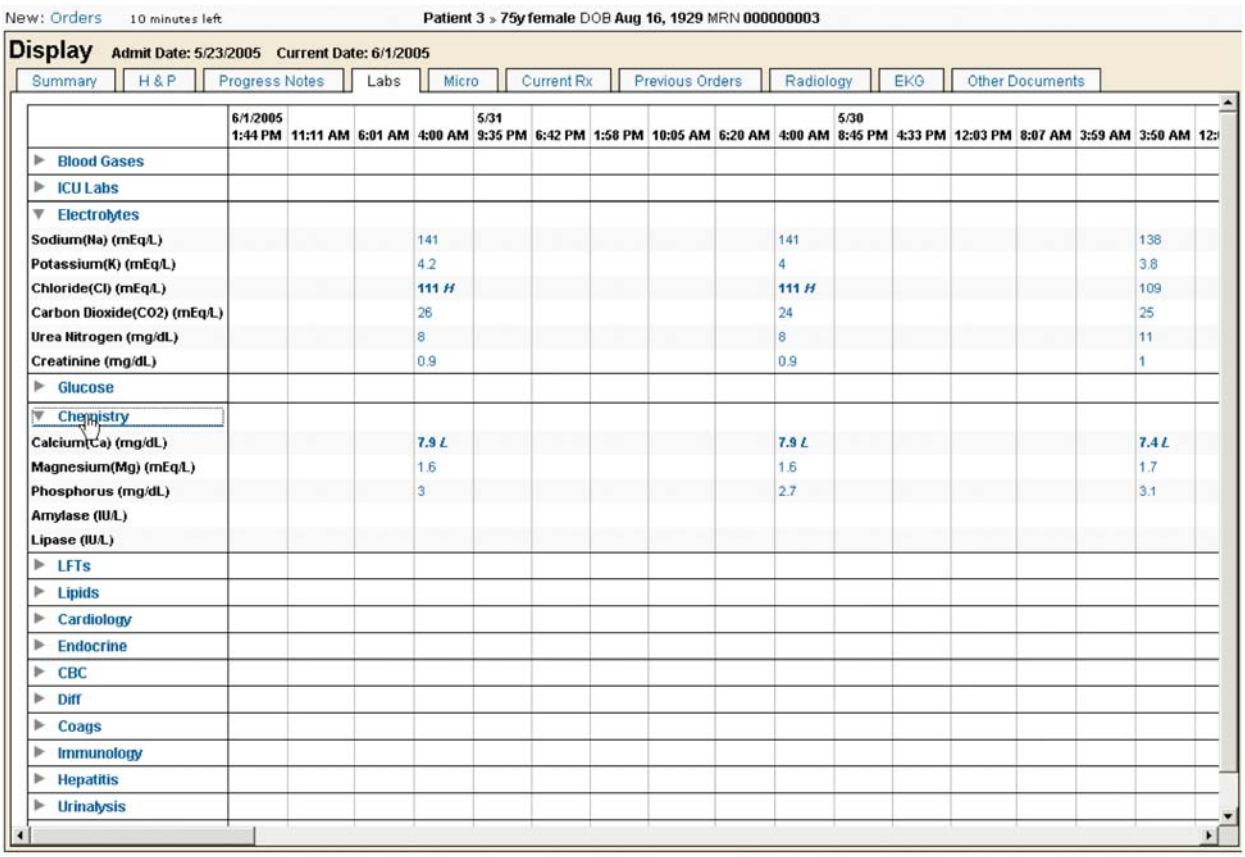


Figure 26: Screenshot of the case presentation software's numerical laboratory data tab ("Labs").

with an optimal length of between seven and nine days, and to generate a problem list and set of orders for the selected stopping point (see Appendix B). A good stopping point was defined as a point at which it would be reasonable to end the case and ask for orders, and at which the time course of changing laboratory values would contribute in some way to the recommended order set. The physicians estimated that 15 minutes would be required to allow sufficient time for reviewing each case and writing orders. Six cases were chosen for use in this study with a range of complexities. Details of the cases are presented below.

5.3.2 Case presentation software

The case presentation software was implemented as a collection of Java Servlets (java.sun.com) generating HTML, with case data stored in a MySQL (www.mysql.com) relational database. The software is intended to be similar in style to how production results review applications organize clinical data. Multiple tabs display case data in source-oriented categories. The leftmost tab, "Summary," (Figure 24) which is open when the software first displays a case, contains a one-sentence summary and problem list. Several document tabs (Figure 25) contain the history and physical exam report, progress notes, consultation reports, radiology reports, and EKG reports.

The Laboratory ("Labs") tab displays laboratory results either in tabular numerical form or with TPOD. The tabular numerical form organizes tests into static categories with results displayed in reverse chronological order, as shown in Figure 26. These categories are similar to those used in production electronic medical records. Disclosure triangles next to each category's name allow data to be hidden or revealed. Abnormal results are displayed in boldface and annotated as ***H*** (High) or ***L*** (Low). Clicking on a result causes a pop-up box to appear with the test's normal range and laboratory comments, if any. The TPOD-based display is described in Section 5.2.

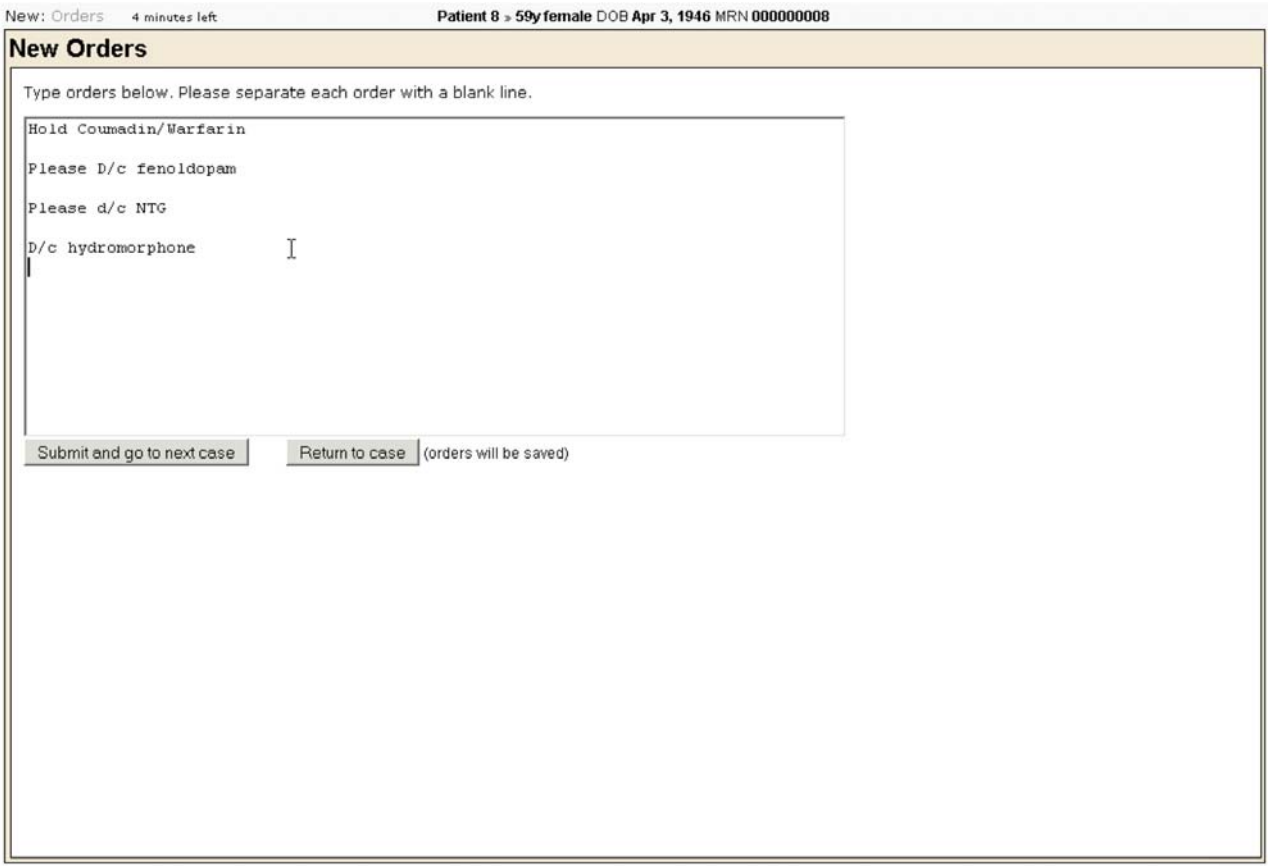


Figure 27: Screenshot of the case presentation software's New Orders screen.

A hyperlink in the upper-left corner of the screen leads users to a New Orders screen, shown in Figure 27, where users can write orders into a text area. Users can move between the New Orders screen and results review without losing previously written orders.

The software was deployed on an Apple Xserve G5 computer (www.apple.com) running JBoss Application Server (www.jboss.org).

Table 2: Data clusters and the cases in which they appear (see case definitions in Table 4 and Table 5). The temporal abstraction primitive (algorithm) for each low-level abstraction is shown in parentheses. Temporal relationships are shown in italics. Data cluster definitions are continued in Table 3.

Name	Cases	Abstraction (algorithm)	Associated data
Unstable glucose	1, 3	4 sequential glucose values with average difference > 50 (variability)	Glycosylated hemoglobin A1c
Renal failure	1, 2, 4, 8	3 of 4 sequential creatinine values > 2.5 (state) *overlapping* 3 out of 4 BUN values > 25 (state)	Na, phosphate
Developing anemia	2	Overlapping declining hematocrit and declining hemoglobin. • Declining hematocrit: 3 sequential values with a decrease >= 2 per day (trend), *immediately followed by* at least 2 out of 3 sequential values < 34.1 (state) • Declining hemoglobin: 3 sequential values with a decrease of >= 0.5 per day (trend), *immediately followed by* at least 2 of 3 sequential values < 11.6 (state)	RBC, MCV, MCH, MCHC, RDW
Supratherapeutic INR	2	2 of 3 sequential INR values > 3 (state)	PT, PTT
Elevated but subtherapeutic INR	2	2 of 3 sequential INR values > 1.25 and < 2 (state)	PT, PTT
Suggest secondary hyperparathyroidism	2	PTH > 65 (state) with the following *within +/- 10 days*: • 2 of 4 sequential phosphate values > 4 (state) • 2 of 4 sequential Ca values < 9 (state) • 3 of 4 sequential creatinine values > 1.4 (state)	Na, K, Cl, CO_2, BUN
Cholestatic liver disease	3	Alk phos > 125 (state) with the following *within +/- 3 days*: GGT > 40 (state) and ALT > 40 (state)	AST, Total bilirubin, Conjugated bilirubin, Delta bilirubin
Inflammation/infection	3	2 of 3 sequential WBC values > 10.6 (state), containing at least 1 bands > 5 (state)	None
Cardiac injury	3	2 sequential troponin values > 0.1 (state)	None

5.3.3 Knowledge base construction

An expert physician who was familiar with TPOD and PROTEMPA reviewed the laboratory data in the cases and created a set of 18 data clusters (textual descriptions are shown in Table 2 and Table 3). Cluster definitions were given to the author, who entered their specifications into TPOD's knowledge base. Of the 18 clusters, 14 specify low-level abstractions and 5 specify temporal pattern abstractions. The low-level abstractions incorporate general temporal abstraction primitives for detecting states, trends, and variability in a data sequence, with arguments specific for particular laboratory tests. The *Developing anemia* cluster's abstraction is illustrated in Figure 28. A total of 26 associated laboratory tests are specified by these clusters.

5.3.4 Study protocol

Internal medicine and family practice residents in the PGY-2 or PGY-3 year were invited to participate for a gift certificate from Amazon.com. Subjects were recruited using flyers posted in Presbyterian, Montefiore, and Shadyside hospitals, a presentation at the Shadyside Hospital Family Practice residency program's morning rounds, and cooperation from the Internal Medicine and Family Practice residency directors in informing residents about this study.

Table 3: Data clusters and the cases in which they appear (see case definitions in Table 4 and Table 5), continued from Table 2. The temporal abstraction primitive (algorithm) for each low-level abstraction is shown in parentheses. Temporal relationships are shown in italics.

Name	Cases	Abstraction (algorithm)	Associated data
Hyponatremia	3, 4	2 of 3 sequential Na values < 136 (state)	
Hypocalcemia	3, 8	2 of 3 sequential Ca values < 8.5 (state)	Na, Phosphate
Hypercalcemia	4	2 of 3 sequential Ca values > 10.5 (state)	Phosphate, PTH, 25-hydroxyvitamin D
Declining calcium	4, 8	3 sequential Ca values with a decrease > 0.5 per day and the last value < 8.5 (trend)	Phosphate, Na
Hypokalemia	4	2 of 3 sequential K values < 3.5 (state)	None
Hypoxia	8	2 sequential pO_2 values < 70 (state)	Arterial pH, O2 sat, pCO2, whole blood lactate
Mixed metabolic/respiratory acidosis	8	2 of 3 sequential pH values < 7.35 (state), *overlapping* 2 of 3 sequential pCO_2 values > 44 (state), *within +/- 12 hours of* 2 of 3 sequential arterial bicarb values < 22 (state)	pH, pO_2, whole blood lactate
INR increasing to supratherapeutic	8	3 sequential INR values with an increase >= 1.0 per day and the last value > 3.0 (trend)	PT, PTT
Hyperkalemia	8	3 sequential K values > 5 (state)	K, Cl, creatinine

The author ran all of the study sessions. Informed consent was obtained (see Appendix C), and instructions were presented verbally according to a prepared script (see Appendix D). Subjects were told that they are covering patients for another colleague, and they are rounding on each patient for the first time. They were told that they had 15 minutes for each case to determine the current status of the patient and write a set of orders for what to do next. In order to introduce time pressure, they were also told that the subject who writes most complete and accurate set of orders in the least amount of time would receive an additional gift certificate. Subjects were asked to "think aloud" as they review the cases and write orders. Instructions and practice in "thinking aloud" were provided. The first two cases (cases 1 and 5, described in Table 4) were used as practice cases for gaining familiarity with the software and the numerical and TPOD laboratory data display styles. The four remaining cases (cases 2, 3, 4, and 8, described in Table 5) were used as test cases. The laboratory data display style for the first test case was chosen randomly, and subsequent cases alternated display styles. Test cases were shown in random order.

Subjects viewed cases on a Windows PC with a 1024x768 LCD display running the Firefox web browser (www.mozilla.org). Subjects were only shown data up to the expert-selected stopping point for each case. Camtasia Studio software (www.techsmith.com) recorded subjects' verbal utterances and on-screen actions and stored them together as an AVI movie file. The case presentation software logged which cases subjects encountered, subjects' navigations from screen to screen, orders, and time spent on each case. Sessions were conducted in a quiet conference room.

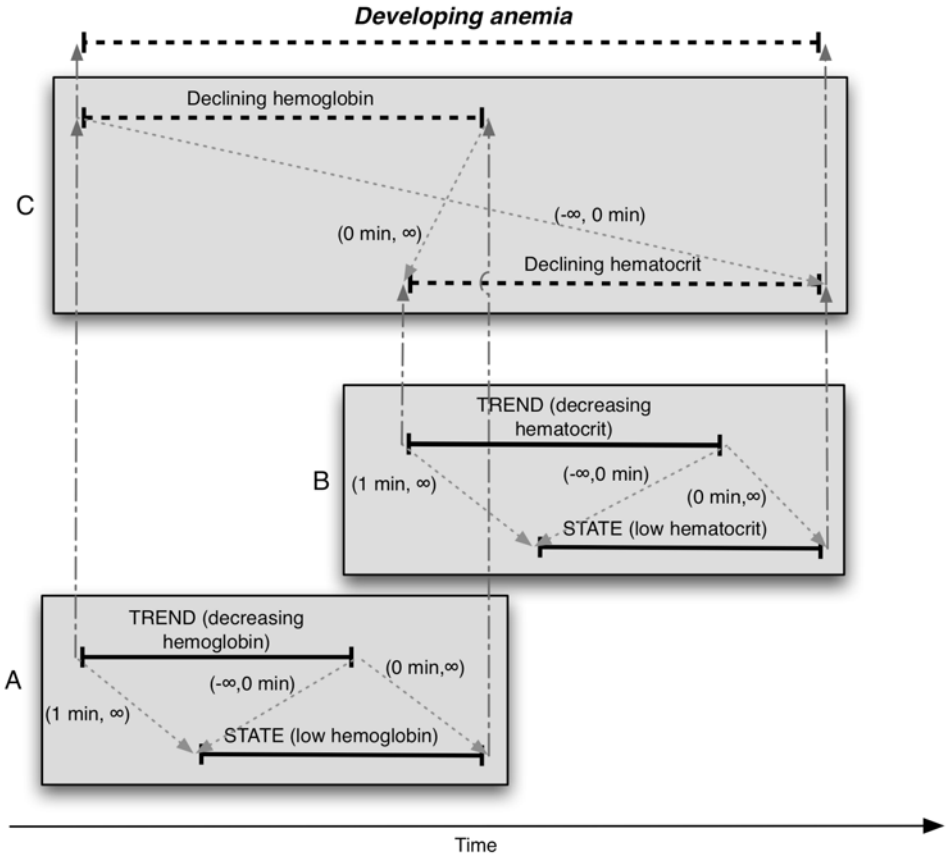


Figure 28: *Developing anemia* temporal pattern abstraction (Table 2). In box A, an *immediately followed by* relationship is specified between *decreasing hemoglobin* and *low hemoglobin* low-level abstractions, which if found, creates a *Declining hemoglobin* abstraction that spans the temporal extent of the component abstractions. Similarly, an *immediately followed by* relationship is specified between *decreasing hemoglobin* and *low hemoglobin* abstractions (box B), which if found, creates a *Declining hematocrit* abstraction. In box C, an *overlapping* relationship is specified between the *Declining hemoglobin* and *Declining hematocrit* abstractions, which if found, creates a *Developing anemia* abstraction. Relationships are specified as described in Section 3.2.2 and Figure 15.

Upon completion of the cases, subjects completed a brief exit interview (see Appendix E) during which they were asked for feedback about the software and about the laboratory data display in particular. Subjects' answers were recorded by Camtasia Studio.

5.3.5 Orders

An expert physician evaluated the orders blinded to the laboratory display style used for each case. Only orders to be carried out in the short term were considered. Orders meant for

Table 4: Descriptions of training cases.

Case #	# days	# of lines	# of orders	# of labs	Active problems
1	4	1641	28	247	• Cirrhosis • History of esophageal varices s/p banding • Acute renal failure with history of chronic renal insufficiency • Pulmonary edema • Pancytopenia • Urinary tract infection • Diabetes • Hypertension • Depression and anxiety • Chronic low back pain
5	3	418	10	122	• Diverticulosis • GERD • Esophageal ulcer • COPD • Mild CHF • Hypothyroidism • Elevated WBC • Anemia

future days and orders for consults were discarded. Doses and operational details of therapy or test drawing were ignored, unless the dose given by the subject indicated a substantially different use of a therapy. The remaining orders were classified as *primary therapeutic* (addresses a patient's active problem list), *secondary therapeutic* (e.g., supportive measures, comfort), *diagnostic* (e.g., chest x-ray), *nursing*, or *other* (e.g., diet, discharge planning, administrative). The nursing and "other" orders were discarded.

The diagnostic and therapeutic orders were each assigned a goal, and these goals were used for subsequent comparisons. For example, the order *Digoxin .5 mg IV now* in a setting of cardiac failure was assigned the goal *inotropic agent*. This approach allowed drug substitutions or different approaches to the same clinical problem to be scored as correct orders.

Orders were scored as *concordant* (order/goal is consistent with the therapeutic goals of the gold standard order set), *discordant* (order/goal is in conflict with the goals of the gold standard order set), *omitted* (the subject omitted an order/goal in the gold standard order set), or *added* (the subject's order/goal is not in the gold standard order set).

5.3.6 Verbal protocol coding for analysis 1

The "think-aloud" data were transcribed in conjunction with the video screen captures and logs. For the preliminary analysis, the author coded these transcripts in order to document the purpose for which subjects navigated to the laboratory display as one of:

1. Overview: scanning for new information with no problem- or test-specific purpose.
2. Evidence seeking: confirmation of a specific problem, test result, or test battery.
3. Completeness: scanning for anomalous data to ensure that nothing has been missed.

For navigations to the "Labs" tab that were for the purpose of evidence seeking, the subject's information was additionally recorded (e.g., "Recent surge in white count?").

5.3.7 Verbal protocol coding for analysis 2

For the TPOD evaluation, the transcripts were coded by the author according to a modified version of a previously reported scheme for identifying cognitive steps in decision-making (36). The original scheme is designed for evaluating a data display with a single screen that presents only raw data, and contains 21 types of cognitive steps from three categories. The first category, *information acquisition*, is for coding when subjects scan a display or read data from a display. The other categories, *information combination and evaluation*, are for coding when subjects interpret data in a display, or make a decision with data in a display.

Table 5: Descriptions of test cases.

Case #	# days	# of lines	# of orders	# of labs	Active problems
2	13	1335	53	396	• S/p I&D of right elbow • Anemia • History of hypercoagulable state • History of CAD and CHF • ESRD • Hypertension • Lupus ? stable/inactive • Recent septic right shoulder joint
3	10	2232	83	472	• CHF • Morbid obesity • Type 2 diabetes • Hypertension • History of atrial fibrillation • Ascending cholangitis • Right upper lobe pneumonia • Deconditioning • Stage II decubitus ulcer buttock
4	8	865	45	317	• Fever • Urinary tract infection • Hypertension • Chronic Renal Failure • Diabetes Type 2 • Malnutrition • Deconditioning
8	6	857	53	238	• ARF • Hypercarbic Respiratory Failure • Rib Pain secondary to fx • PVD • Hx TIA • Spinal Stenosis • COPD • S/p AICD • CAD s/p CABG • Hx DVT

Several changes were made to this scheme (shown in Table 6 and Table 7) in order to adapt it to evaluating a multi-screen data display that presents both raw data and interpreted concepts (patterns). A new category of screen navigation codes was added for associating a cognitive step with a screen or tab (Table 6). New codes were added to the information combination and evaluation category (Table 7) for coding when a subject reads a concept from the display, and for differentiating between cognitive steps taken while viewing the numerical laboratory data display and cognitive steps taken while viewing the TPOD-based display. Codes were created for reading a concept from a laboratory display (RCG, RCN, RRCN and RRCG in Table 7), and for inferring a concept from a laboratory display (ICG and ICN in Table 7), in order to compare the total number of concepts read or inferred between the TPOD-based laboratory display (in which concepts may be read from the display or inferred from raw data) and a tabular numerical laboratory display (in which concepts must mostly be inferred). Finally, sub-codes were created for describing cognitive steps identified with an information combination and evaluation code in more detail (Table 7). See Appendix F for an example of a coded transcript.

5.3.8 Statistical analysis

In Analysis 1, subjects' navigations to the tabular numerical laboratory data display were compared by purpose (overview, evidence seeking, or completeness). Results were fit to a marginal Poisson regression model using General Estimating Equation (GEE) (149) in order to compare counts in the setting of small sample size. Comparisons of evidence seeking versus overview and evidence seeking versus completeness were performed using an alpha level of .05.

Table 6: Updated coding scheme for think-aloud protocols (continued in Table 7).

Code	Explanation
Screen navigation	
GS-S	Go to screen – Summary
GS-H	Go to screen – H&P
GS-PN	Go to screen – Progress Notes
GS-L	Go to screen – Labs
GS-M	Go to screen – Micro
GS-C	Go to screen – Current Rx
GS-PO	Go to screen – Previous Orders
GS-R	Go to screen – Radiology
GS-E	Go to screen – EKG
GS-O	Go to screen – Other Documents
GS-NO	Go to screen – New Order
Information acquisition	
RE	Read a value from the display
RC	Read a concept from clinical data
BR	Browse the display (followed by SE or GS-NO)
CBR	Complete browsing
SE	Search against a criterion (followed by CSE or ISE)
CSE	Complete search
ISE	Interrupt search
RV	Review
CRV	Complete review
RS	Refer to a search mentioned previously
AL	Refer to an alternative mentioned previously
AT	Refer to an attribute mentioned previously
VA	Refer to a value mentioned previously
EV	Refer to an evaluation of an alternative mentioned previously
RRC	Refer to a read concept from clinical data

Analysis 2 employed a one-factor within-subject design in which the factor was display style. This analysis compared 1) the total time spent on each case; 2) the total time spent viewing the laboratory display; 3) the proportion of information acquisition versus information combination and evaluation steps; 4) the number of cognitive steps while viewing the laboratory display; 5) the number of navigations to each screen; 6) the proportion of orders that were *concordant*; and 7) the raw counts of *concordant*, *discordant*, *omitted*, and *added* orders. The results of analyses 1, 2, 3, 5, and 7 were fit to a marginal Poisson regression model using General Estimating Equation (GEE) (149). The results of analysis 4 and 6 were each evaluated using a paired t-test (36) with an arcsine transformation to stabilize the variance (36, 150). All statistical tests used an alpha level of .0045 (Bonferroni correction with 11 comparisons for a total alpha level of .05).

5.4 RESULTS

Seven subjects completed a total of twenty-eight test cases. The introductory script and practice cases took between 35 and 40 minutes to complete. The test cases and exit interviews resulted in 18 hours and 12 minutes of audio and video. One subject ran out of time in one case with the TPOD display style, but otherwise subjects completed all cases. Subjects viewed two test cases in each display style.

Table 7: Updated coding scheme for think-aloud protocols (continued from Table 6).

Code	Explanation
Information combination and evaluation	
IC	Infer a concept (not from laboratory displays)
ICN	Infer a concept from the numerical laboratory display
RCG	Read a concept from the graphical laboratory display (explicitly identified)
ICG	Infer a concept from the graphical laboratory display (not explicitly identified)
RCP	Read about a care plan
RRCP	Refer to read care plan
RIC	Refer to a concept (not from laboratory displays)
RCN	Refer to a concept from the numerical laboratory display
RRCN	Refer to a concept from the numerical laboratory display
RRCG	Refer to a concept from the graphical laboratory display
RICG	Refer to an inferred concept from the graphical laboratory display
AC	Compare diagnostic decision alternatives
MAC	Compare management decision alternatives
CA	Cancel (ignore) an insignificant value or attribute
EMA	Express evaluation of a management alternative
EDA	Express evaluation of a diagnostic alternative
IC, ICN, RCG, and ICG may have one of the following sub-codes:	
-DI	Refer to the difference between two values
-CN	Count occurrences of some value
-AD	Refer to the sum of two or more values
-QO	Refer to the quotient of two values
-IN	Refer to inconsistent values for two or more attributes
-OC	Otherwise compare two values
-TC	Note a time course
-T	Note a trend
-MV	Otherwise compare multiple values
-ST	Refer to a cutoff or standard value for an attribute
-STM	Refer to a cutoff or standard value for an attribute (over multiple values)
-WT	Refer to the weight or importance given to an attribute
-PS	Refer to patient-specific cutoff or standard value for an attribute
-PSM	Refer to patient-specific cutoff or standard value for an attribute (over multiple values)
-TR	Refer to a trade-off between values or attributes, or to a compensatory combination of values or attributes
-HV	Hypothesize about value of an unseen attribute
-RV	Refer to a value
Other	
QD	Question why previous decision was made
OT	Other (anything else)

5.4.1 Results of analysis 1

In cases with the tabular numerical laboratory display, subjects navigated to the "Labs" tab 74 times. Subjects' purpose for navigating there could be determined in all cases. Repeated-measures analysis revealed that subjects navigated to the laboratory display primarily for evidence seeking (see Figure 29), and the difference in terms of mean count of navigations per subject was statistically significant compared to both overview ($p = 0.010$) and completeness ($p < 0.001$).

Subjects navigated to the tabular numerical laboratory display for evidence seeking 41 times (55% of total navigations to the "Labs" tab). Subjects either explicitly stated their information need (e.g., "Why were they thinking C. diff when no diarrhea and no micro reported?") or the need was implicit in statements they made and/or the test results they reviewed. Table 8 shows an example of an implicit information need "Confirm high INR (while on coumadin)."

Four categories of information needs were identified: evidence of a problem (e.g., "Dehydration?"), the results of specific tests (e.g., "Recent surge in white count?"), information to help decide whether to change an order (e.g., "Is K-dur needed?"), or information about tests related to an organ system (e.g., "Endocrine"). Table 9 shows the percentage break down of information needs by these categories.

Only one subject navigated to the Labs for seeking information about an organ system. This individual read the Assessment and Plan sections of the progress notes, which were generally organized by organ system, and then scanned the labs by organ system. Thus, it appears that this subject was seeking evidence in support of prior physicians' assessments and plans.

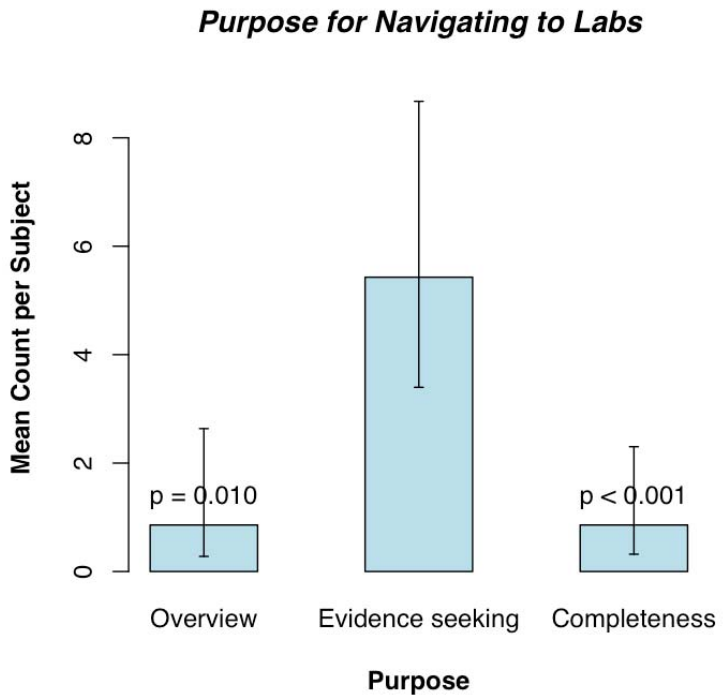


Figure 29: Analysis of purpose for navigating to the tabular numerical laboratory data display (error bars calculated using a Poisson distribution).

Table 8: Implicit statement of an information need.

Time	Actions and verbalizations
5:10	Action*: navigate to Current Rx tab.*
5:11	Verbalization*: "Okay, so her current meds that she's on. She's on coumadin."*
5:16	Action*: navigate to Labs tab.*
5:17	Verbalization*: "I thought her INR was a little bit high. I'm not sure why they started that again."*
5:19	Action*: open Coags section of labs.*
5:20	Verbalization*: "Yeah, coumadin should be stopped."*

Otherwise, subjects exhibited a complex method of examining labs. Of the 41 times that subjects navigated to the laboratory data for evidence seeking, they reviewed additional data unrelated to the stated information goal 9 times (22%). In the excerpt shown in Table 10, the subject navigated to the Labs tab to determine why previous physicians were considering the diagnosis of *C. difficile* when there was no diarrhea and no microbiology results. The subject first reviewed relevant information (CBC and Diff), but then continued reviewing labs down the page (see Figure 26 for section order). The subject reviewed Coags, noted with surprise that a Hepatitis panel had been ordered, reviewed the patient's Urinalysis, and noted again with surprise that Urine Tox had been ordered. Subjects' use of the laboratory data appeared to be influenced by the order in which the sections were presented. In all subjects, 9 laboratory data sections were reviewed in which the subject expressed surprise that those tests had been ordered, for an average of 0.64 unexpected findings per case.

5.4.2 Results of analysis 2

Analysis 2 concerned all twenty-eight test cases.

Time required to review cases

Subjects used the entire 15 minutes for half of the cases. Results are shown in Table 11. There was no statistically significant difference between display styles with respect to the time spent on each case ($p = .70$), or with respect to the time spent viewing the laboratory display ($p = .89$).

Table 9: Categories of information needs.

Confirmation category	Count (percent)
Confirming a problem	19 (46%)
Results of specific tests	14 (34%)
Determining whether to change an order	6 (15%)
Tests related to an organ system	2 (5%)

Table 10: Discovery of an unexpected finding.

Time	Actions and verbalizations
11:46	Action*: navigate to labs*
11:47	Verbalization*: "Let us see what we have here. Why were they thinking C. diff if there was no diarrhea? And let us see."*
11:58	Action*: open CBC section of labs.*
11:58	Verbalization*: "The CBC is not high. I don't think … WBC is normal. I don't think they should think about that."*
12:04	Action*: open Diff section.*
12:05	Verbalization*: "Differential… I don't see anything here. It is all normal."*
12:14	Action*: open Coags section.*
12:14	Verbalization*: "Coags is fine."*
12:16	Action*: open Hepatitis section.*
12:19	Verbalization*: "Was a hepatitis panel done on her? The items are all negative."*
12:25	Action*: open Urinalysis section.*
12:25	Verbalization*: "Okay, urinalysis. Let's see what her urinalysis that was initially done…I'm not going to chase that at this time."*
12:50	Action*: open Urine Tox section.*
12:51	Verbalization*: "Was a U-tox done? I'm not sure (why)…"*

Cognitive steps

There was no statistically significant difference between display styles with respect to the proportion of information acquisition to information combination and evaluation steps, either overall or just while viewing the laboratory display (see Table 11). However, subjects made more cognitive steps for reading or inferring concepts while using the TPOD-based laboratory display (codes ICG, RCG, and RRCG in Table 7) than while using the numerical display (codes ICN, RCN, and RRCN in Table 7), and the difference was statistically significant ($p < .001$, see Figure 30 and Table 11).

Log analysis

Subjects navigated to the laboratory display more frequently in cases with the numerical display style than in cases with TPOD (shown in Table 11), and while the difference was statistically significant at the .05 level ($p = .016$), it was not significant after Bonferroni correction.

Orders

Subjects wrote slightly more *added* orders in cases with the TPOD-based display than in cases with the numerical laboratory display (see Table 12), and while the difference was statistically significant at the .05 level ($p = .026$), it was not significant after Bonferroni correction. There was no statistically significant difference between display styles with respect to the number of *concordant*, *discordant*, or *omitted* orders, or with respect to the proportion of orders that were *concordant* (see Table 12). Similar results were obtained when just considering each subject's therapeutic orders.

Table 11: Within-subject comparison of the numerical versus graphical laboratory data displays, overall and by display style.

Measure per display style within subject	Overall (SD)	Graphical (SD)	Tabular (SD)	*p*
Total time	13.27 min (.47)	13.40 min (.47)	13.14 min (.47)	.70
Time viewing labs	2.70 min (.21)	2.67 min (.21)	2.73 min (.21)	.89
# cognitive steps	158.89 (12.61)	162.14 (12.73)	155.64 (12.48)	.49
Proportion of steps for acquisition	0.69 (.10)	0.69 (.10)	0.67 (.10)	.53
# navigations to laboratory data display	10.57 (3.25)	7.57 (2.75)	13.57 (3.68)	.016
# of concepts read or inferred	26.64 (5.16)	30.86 (5.56)	22.43 (4.74)	< .001

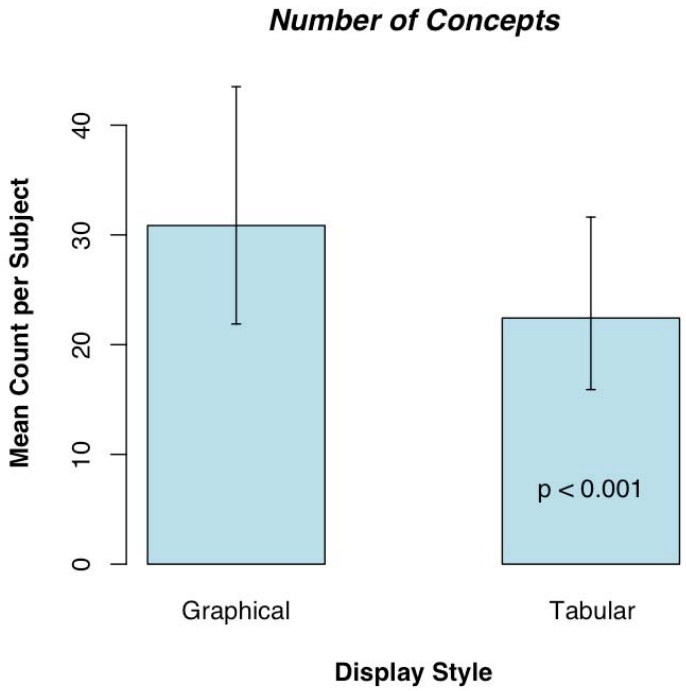


Figure 30: Plot of mean number of times subjects read or inferred a concept from TPOD-based display of laboratory data versus the tabular numerical display (error bars calculated using a Poisson distribution).

User reactions

Subjects commented about the case presentation software and laboratory displays during case review. Five subjects verbalized which laboratory display style they preferred: three preferred the TPOD-based display, and two preferred the numerical display.

One subject was particularly enthusiastic about the TPOD-based display, and commented while using the display:

> This is the one I like. See, that — there's hypernatremia — that's common in patients like that. Hypokalemia. Patient has renal failure. I'm not going to push any potassium, it's not that low. 3.4 High BUN, and creatinine has been high. Associated with ... I like this stuff, I like this display a lot. It tells me that the sodium is low…

Another subject was particularly unenthusiastic about the TPOD-based display, and commented while using the display, "I have no idea what this means. Oh, I see. This is really very non-helpful. I mean, I don't think we need that pointed out for us but I could be wrong."

Two subjects expressed discomfort with using the downward and upward pointing arrows of the TPOD-based display's compressed graphs (see Figure 22) in order to determine if results were high, low, or normal. One subject commented, "Thyroid studies — apparently based on glancing for arrows — is [*sic*] negative." The other commented, "I just rely on this black dot rather than (the) number."

Exit interview

Three subjects expressed a preference for the TPOD-based display, three subjects expressed a preference for the numerical display, and one was undecided. Five subjects

Table 12: Within-subject comparison of orders with the numerical versus graphical laboratory data displays, overall and by display style.

Order categories per display style within subject	Overall (SD)	Graphical (SD)	Tabular (SD)	*p*
Concordant[1]	15.36 (3.92)	15.14 (3.89)	15.57 (3.95)	.71
Discordant[2]	1.07 (1.03)	1.14 (1.07)	1.00 (1.00)	.87
Omitted[3]	9.14 (3.02)	8.71 (2.95)	9.57 (3.09)	.58
Added[4]	6.57 (2.56)	7.86 (2.80)	5.29 (2.30)	.026
Ratio of concordant to inaccurate	1.56 (.13)	1.53 (.13)	1.59 (.13)	.51

[1]Order in subject list consistent with order in gold standard list.
[2]Order in subject list inconsistent with order in gold standard list.
[3]Order in gold standard list and but not in subject list.
[4]Order in subject list but not in gold standard list.

expressed a preference both in the exit interview and during case review, and their preferences were consistent.

Subjects who preferred TPOD, when asked why, responded:

- It helps focus on pertinent laboratory values and determine when in the patient's time course certain events occurred (n=2).
- Good for when a patient has a trend (n=1).
- Colors were "more exciting" (n=1).
- Easier to find data (n=1).
- Tabular display only shows you a few days of data on a screen (n=2).

Subjects who preferred the numerical display, when asked why, responded:

- Can see the TPOD-based display's potential value but need more time with it to decide (n=2).
- The TPOD-based display would be more useful for long hospitalizations (n=1).

- Desired to "see the numbers" (n=3).
- The TPOD-based display is not needed (n=2).
- Concerned that they were just looking for the upward-pointing and downward-pointing triangles rather than looking for the numbers (n=1).
- The graphical display would be more useful for educational purposes (n=2).
- The TPOD-based display would be more useful in an ICU setting (n=1).
- Not trained to analyze the data the way it is presented with the TPOD-based display (n=1).

5.5 DISCUSSION

Existing production electronic medical record displays do not optimally support problem solving in patient care. The organization, form, and sequence of these displays are the same regardless of the patient's underlying problems (see Section 2.2). Displays that identify patterns in time sequences of patient data that are related to active problems and processes might help physicians to make timely and accurate decisions. No studies have evaluated the effects of these display features on clinical decision-making in the context of real patient cases.

This study evaluated physicians' data acquisition behaviors while using a standard tabular laboratory data display during case review, and compared the effects of a process-oriented display of laboratory data (TPOD) on decision-making by physicians. Analysis 1 characterized the information needs that laboratory displays satisfy, and the data acquisition behaviors that physicians use to review data in laboratory displays. The results complement prior research suggesting that physician-written notes are often used for overview (93), in that subjects

more often navigated to the laboratory data display for evidence seeking. When subjects navigated to the laboratory display for evidence seeking, they were primarily interested in known problems, but they also frequently sought information related to other kinds of concepts. This finding suggests that there might be benefits to decision-making for displays that support data acquisition using a wide variety of concepts, not just by problem.

The organization of the Labs tab in the numerical display (see Figure 26) allowed discovery of additional data and unexpected results. It is unclear whether physicians consciously scan for anomalous findings, or if the organization of the display subconsciously encourages it. Regardless, discovery of unexpected results might provide insight into prior physicians' thought processes, even if those results are negative (e.g., a negative Hepatitis panel). A potential challenge with designing concept-oriented results review displays may be how to make unexpected results that are not part of any known problem or concept easily discoverable.

A limitation of Analysis 1 is relatively small counts of navigations to the laboratory data display. The schema for coding navigations (see Section 5.3.6) appears to have face validity, as it is consistent with descriptions of data acquisition behaviors in the literature. However, it would benefit from reliability testing with additional coders.

These results may inform the design of novel data displays, such as TPOD, by highlighting the relationship between display organization, data acquisition behaviors, and information needs. Novel displays will likely need to support the same information needs as existing displays, although they may do so by supporting different data acquisition behaviors.

TPOD is a novel data visualization strategy and software tool that dynamically assembles evidence of clinical processes, and aggregates data that may be useful for drawing conclusions about a patient's status. TPOD adapts its organization, form, and sequence entirely based upon

automated analysis of the underlying data. In contrast, existing concept- or problem-oriented displays rely on an external problem list to determine a display's organization (103), or require the user to specify what concepts or problems are of interest (28). Unlike existing systems, TPOD supports "intermediate" concepts (see Section 2.3) such as patterns in laboratory results, which may be integral to identifying disease processes but are not typically part of diagnostic problem lists. Thus, TPOD supports visualization of a wider variety of clinical concepts than existing systems.

The rank field specified by each data cluster assigns a priority to each cluster, and is intended to help physicians identify important patient-specific patterns that require their attention. Ranking would be expected to help physicians obtain overviews of laboratory data and discover unexpected data interpretations. Since Analysis 1 was not carried out using the TPOD-based laboratory display, it is unknown at this time whether subjects actually used data clusters to satisfy those needs. The ranking system employed in this study is preliminary. A complete mechanism for assigning priority to data clusters might take into account the patient's underlying conditions, what data the user has already reviewed, and the user's role. Such a mechanism might also involve further changes to how clusters of different priority are displayed besides hiding or revealing, such as displaying data at different levels of detail, or changing the size of graphics.

TPOD's use of graphics is designed to facilitate data interpretation in several ways. Relationships between test results are more easily identified. Data is displayed more compactly, so limited horizontal scrolling is required in order to see data from the past. Abnormal test results can be easily spotted by color and arrow direction without having to manually compare each test result to its normal range. While traditional laboratory data displays also annotate individual test

results to indicate abnormal results, several subjects reported in the exit interview that TPOD made it easier to identify abnormal results and trends.

The organization of data in TPOD is less predictable than with traditional laboratory displays that organize tests into static groups. User interface design guidelines frequently articulate the desirability of user interface consistency in a display's organization (e.g., information and actions should always be in the same place on the screen). On the other hand, slavish adherence to consistency may be a detriment to usability when consistency is given priority over users' work environments (151). TPOD is designed with the latter approach in mind. While TPOD does dynamically reorganize data, processes are all visualized using the same basic form and internal organization, and users interact with each process using the same user interface techniques (e.g., vertical magnification, sliding the mouse over graphs, pop-up boxes for details).

Subjects obtained more concepts from the TPOD-based display, as measured by verbal protocol analysis of subjects' cognitive steps. This result suggests that the TPOD based display's visualizations were readable despite limited training time. However, no statistically significant differences were found in decision-making speed, the proportion of information acquisition cognitive steps, or in measures of order quality.

The lack of a difference in measures other than concepts might be due to an insufficient number of subjects. There is little precedent in the literature for determining the power of studies such as this. A minimum sample size of 12 subjects was originally aimed for, with the intent of recruiting additional subjects if needed. Only 7 subjects could be recruited. Given the results of Analysis 2, a power of .9, and a study-wide Type I error controlled at the .05 level, a sample size of 32 subjects would have been needed.

An alternative explanation is the fact that the tabs in the two display conditions were identical except for the Labs tab. Subjects spent most of their time viewing tabs other than the laboratory displays. Furthermore, at least in the cases with the numerical display, subjects obtained laboratory results from clinical documents as well as from the laboratory display. Constraining the study, such as by completely removing laboratory results from the clinical documents, might have increased the likelihood of seeing a difference in decision-making. However, such constraints would have forced subjects to use the laboratory display for obtaining an overview of the case, even though physicians do not appear to use the laboratory display for that purpose in reality.

Lack of familiarity with the TPOD-based display might have also contributed to the lack of difference. More training time might have reduced the number of subjects who felt this way, but it was felt that two hours was the maximum session length that subjects could have reasonably been expected to undertake.

Requiring subjects to perform an action in order to see the numbers might have also inhibited subjects' performance with TPOD, given that several subjects complained in the exit interview about not being able to "see the numbers" easily. Given that two subjects expressed uncertainty while reviewing cases about relying on the upward and downward pointing arrows instead of looking at the numbers, and that subjects expressed a lack of familiarity with TPOD in the exit interview, subjects might have not seen this as a problem had they been more familiar with TPOD. Practically, displaying all of the numbers alongside the graphs would create too much screen clutter, potentially obscuring the trends and patterns that TPOD aims to visualize. An alternative approach might be to display the numbers that clinicians most often use (e.g., the most recent result for each test, the results of admission labs).

Verbal protocol analysis was a useful method of measuring cognitive steps in this study, although it appears to be limited in its ability to detect data acquisition steps for displays with large volumes of quantitative data. While subjects did read aloud a large proportion of the data that they appeared to be reviewing in the document displays, they tended not to read aloud a laboratory result unless it was particularly interesting. Verbal protocol analysis may need to be supplemented with other techniques such as eye tracking (152), in which a camera records subjects' eye movements as they look at a display, in order to obtain an accurate count of the number of data elements being read in the laboratory displays.

Despite issues regarding the ability of verbal protocol analysis to measure information acquisition steps in the laboratory displays, the proportion of information acquisition versus information combination and evaluation steps obtained in this study is similar to those reported by the authors of the original coding scheme (a 2:1 ratio of information acquisition to information combination and evaluation steps) (36). This similarity of results lends validity to the modified coding scheme's ability to detect the desired cognitive steps. Nonetheless, the modified scheme would benefit from reliability testing with additional coders.

A challenging use case, results review of unfamiliar cases, and relatively long and complex inpatient cases were purposefully chosen in order to stress the capabilities of the data displays and induce time pressure. Previous studies in other domains suggest that graphical displays show benefit especially in time pressured scenarios (see Section 2.2.3). Other use cases may exhibit variations in information needs and data acquisition behaviors with which clinicians browse components of the record.

5.6 CONCLUSIONS

This chapter describes a process-oriented data display, TPOD, and a comparative study of a laboratory display based on TPOD versus a tabular numerical laboratory display in the context of real patient cases. A preliminary analysis of the tabular display suggests that physicians use laboratory displays for multiple information needs while reviewing unfamiliar patient cases. Evaluations of clinical displays that only provide access to a part of the medical record out of context may bias results by forcing subjects to use a display for information needs that they would normally satisfy in other parts of the medical record. These results need to be confirmed in a larger study for other laboratory data display styles, for displays of other kinds of clinical data, and for different decision-making tasks.

TPOD successfully visualized a set of complex temporal and mathematical patterns in clinical laboratory results. While no differences in decision-making quality and speed were found, subjects acquired more concepts from the TPOD-based display. The increased communication of concepts is consistent with improved decision-making. Why improved decision-making was not detected is unclear. Nonetheless, comparative evaluation of novel data displays in context appears to be a useful methodology that provides insights into physicians' preferences, the process of clinical decision-making by physicians, and display usability. More subjects, improved ways of detecting cognitive steps in complex data displays, and more training time on novel displays may be needed in order to detect differences in decision-making speed and accuracy in these kinds of studies.

6.0 DISCUSSION

The preceding chapters describe the development and evaluation of a process-oriented approach to specifying, identifying, and visualizing temporal and mathematical patterns in clinical data sequences. This chapter summarizes the main findings and contributions to biomedical informatics, and describes potential future work.

6.1 SUMMARY

Identifying relationships between the values and timestamps of clinical data sequences is important for correct interpretation of the data in a diverse range of clinical tasks (Section 2.1.1), but is hampered by clinician information overload (Chapter 2.0). Previous approaches to reducing information overload include automated interpretation (Section 2.1) and improved visualization (Section 2.2) of time-stamped patient data. A few software systems have combined temporal and mathematical pattern detection with pattern visualization (Section 2.2.4), but these systems have been developed and evaluated primarily for results review. Evaluations of such displays have involved limited data sets, and subjects in these studies were not asked to use these displays in the context of reviewing and making decisions about reasonably complete inpatient cases.

The preceding chapters introduce an integrated process-oriented approach to reducing information overload. The data processing component of the process-oriented approach,

Table 13: Patterns and processes identified by PROTEMPA, and the temporal abstraction primitives that were required to identify these patterns.

Process type	Pattern examples	Temporal abstraction primitives	Described in
Disease states	HELLP syndrome, cholestatic liver disease	State, trend	Figure 18 and Table 2
Disease progression and outcomes	HELLP syndrome disease severity and outcome	State, trend, minimum	Table 1
Therapeutic response	Electrolyte status, coagulation status, renal function	State, trend, variability	Table 2 and Table 3

PROTEMPA (Chapter 3.0), extends temporal abstraction (Section 2.1.2) with a flexible framework for defining, storing, executing, and sharing a library of general-purpose or task-specific temporal abstraction primitives (Section 3.2.2). The contribution of this framework is to facilitate the use of temporal abstraction in domains besides data summarization and monitoring. PROTEMPA has successfully identified such patterns in patient care and clinical research tasks (Chapters 4.0 and 5.0). These patterns, shown in Table 13, are not all supported by the standard set of temporal abstraction primitives (e.g., state, trend, rate) provided by existing temporal abstraction systems. An application of PROTEMPA for identifying populations of patients for medical process improvement has been partially evaluated and is described in Future Directions below.

The data visualization component of the process-oriented approach, TPOD (described in Section 5.2), has been developed and evaluated for its ability to enhance decision-making during results review (Section 5.3). TPOD successfully visualized patterns identified by PROTEMPA for prospective analysis of individual patients, and in an evaluation of the effects of a TPOD-based laboratory display on decision-making speed and accuracy as compared with a tabular numerical laboratory display in the context of complete patient cases, subjects read or inferred

more concepts from the TPOD-based display than from the tabular display, although no effects on decision-making speed or accuracy were found.

The main contribution of TPOD is as a novel adaptive concept-oriented display, in which concepts are only visualized if the patient's data satisfies certain constraints. The comparative evaluation of TPOD is also a contribution as it extends cognitive data display evaluation methods (Section 2.2.5) in order to compare a novel data display to a control display in the context of complete patient cases. This kind of evaluation may be useful for determining a novel display's effects on the process and outcome of clinical decision-making, and for defining how clinicians use clinical data displays in context.

Previous authors have described a method of evaluating decision-support systems that starts with a laboratory evaluation of the system's technical performance, and concludes with a field trial in which the system is evaluated for its effects on clinical practice and outcomes (70, 153). The evaluation methodology described in Chapter 5.0 may serve as an intermediate step between the performance evaluation and the final evaluation in the field. In particular, it may identify usability issues and user reactions that, if addressed, may increase the likelihood of a successful field trial, and improve the chances that users will accept the system when it is deployed in production.

6.2 FUTURE DIRECTIONS

Several areas of future work are planned to demonstrate the applicability of PROTEMPA and TPOD to a diverse set of clinical tasks, in addition to the future work described in the Discussion sections of Chapters 3.0, 4.0, and 5.0.

6.2.1 Clinical database query

Chapter 4.0 describes prototype software for retrieving patients of interest from clinical data repositories who are described by mathematical and temporal patterns in data sequences. A complete retrospective data retrieval sequence would initially identify a subset of patients appropriate for processing from a data repository using coded, demographic or other data with standard query tools. PROTEMPA would extract corresponding data from the repository as necessary and create a database of found patients and associated temporal intervals. This database would be passed to a final stage that categorizes the patients based on the found intervals and query requirements, and presents the data in a form that is appropriate for users. A user interface that provides an integrated workflow would be implemented to construct queries and review the patient populations that are found.

6.2.2 Knowledge discovery

Identification of temporal and mathematical patterns in large data sets may enhance knowledge discovery. Machine learning and data mining (154) techniques identify relationships between elements in these data sets that may improve understanding of the incidence and progression of disease as well as response to therapy (155). However, conventional methods are not designed to recognize relationships between data elements as such and thus may regard a temporal sequence as an unrelated aggregate of individual data elements rather than an entity with intrinsic clinical meaning. Identification of abstractions within large data sets using PROTEMPA may reveal these clinically-meaningful temporal patterns as distinct entities and thus render them accessible for unsupervised and supervised machine learning approaches (87).

6.2.3 Medical process improvement

An improved capability to identify errors in patient care processes is recognized (156) as important for enhancing patient safety (157). In the clinical laboratory, bedside or point-of-care testing (158) is a focus for process improvement. Bedside testing has become a routine part of managing hospitalized patients. Because the bedside is a less controlled environment than the central laboratory, bedside testing may be more susceptible to errors due to operator variability or equipment malfunctions (159, 160). Error detection may require specialized calculations on time sequences of laboratory results, and comparisons between the values of multiple tests. Identifying these patterns by hand can be challenging because the volume of bedside test results is large, typical laboratory displays do not facilitate detection of relationships between multiple test results (see Section 2.3), and existing software systems are not designed to correlate bedside monitoring data with results from the central laboratory. These challenges are similar to those encountered when interpreting data for diagnosis and patient monitoring, as discussed in Chapters 4.0 and 5.0.

Preliminary work has been done to extend PROTEMPA and TPOD for identifying and visualizing patient care process problems in bedside testing. Specific goals for monitoring bedside testing for process problems include 1) ensuring that new operators are correctly administering the test, 2) detecting equipment malfunction, and 3) facilitating follow-up of problem patterns (e.g., equipment maintenance, in-service training for equipment operators, or process re-design).

Table 14: Abstraction definitions implementing the American Diabetes Association (ADA) guidelines for bedside glucose testing, with the temporal abstraction primitive (algorithm) for each low-level abstraction shown in parentheses.

Name	Abstraction (algorithm)
Lab and POC glucose within 30 minutes	Lab and bedside glucose that are within 30 minutes of each other (delta check)
Inconsistent Lab/POC glucose (Lab glucose > 70)	Lab and bedside glucose that are within 30 minutes of each other, the lab glucose is > 70, and the first and last values are > 10% apart (delta check)
Inconsistent Lab/POC glucose (Lab glucose <= 70)	Lab and bedside glucose that are within 30 minutes of each other, the lab glucose is <= 70, and the first and last values are > 15 apart (delta check)
More than 1 POC glucose within 10 minutes	> 1 bedside glucose within 10 minutes (frequency)
Inconsistent POC glucose > 70	> 1 bedside glucose within 10 minutes in which the first value is > 70, and the first and last values are > 10% apart (delta check)
Inconsistent POC glucose <= 70	> 1 bedside glucose within 10 minutes in which the first value is <= 70, and the first and last values are > 15 apart (delta check)

The American Diabetes Association (ADA) provides guidelines for ensuring that bedside testing of blood glucose has similar precision and accuracy to central laboratory testing (161). These guidelines were implemented as PROTEMPA abstractions for detecting intervals of 1) excessive variability in sequential bedside glucose results (a test for precision), and 2) excessive variability in contemporaneous bedside glucose and laboratory glucose results (a test for accuracy) (see Table 14). Additional low-level abstractions were created for detecting 1) all bedside glucose results with another contemporaneous bedside glucose result (*More than 1 POC glucose within 10 minutes*), and 2) all bedside glucose results with a contemporaneous laboratory glucose result (*Lab and POC glucose within 30 minutes*) (Table 14). The latter two clusters provide a baseline from which to determine the proportion of a patient's contemporaneous bedside and laboratory glucose results that have excessive variability, and the proportion of a patient's contemporaneous bedside glucose results that have excessive variability. These

Table 15: Intervals found in 28 days of bedside glucose and laboratory glucose data (see Table 14 for abstraction definitions). POC = point-of-care (bedside).

Group	Category	Number (percent of total)	% of Total
By patient	Total patients with excessive variability in lab/bedside glucose results[1]	281	5.4
	Total patients with excessive variability in bedside glucose results[2]	89	1.7
	Total patients with intervals	*319*	*6.1*
	Total patients	*5219*	*100*
	Average number of new patients identified per day	*11.4*	—
By abstraction	Instances of excessive variability in lab/bedside glucose results[3]	506	82.5
	Instances of excessive variability in bedside glucose results[4]	107	17.5
	Total intervals	*613*	*100*

[1]Total number of patients with at least one *Inconsistent Lab/POC glucose (Lab glucose > 70)* or *Inconsistent Lab/POC glucose (Lab glucose <= 70)* interval.
[2]Total number of patients with at least one *Inconsistent POC glucose > 70* or *Inconsistent POC glucose <= 70* interval.
[3]Total number of *Inconsistent Lab/POC glucose (Lab glucose > 70)* and *Inconsistent Lab/POC glucose (Lab glucose <= 70)* intervals.
[4]Total number of *Inconsistent POC glucose > 70* and *Inconsistent POC glucose <= 70* intervals.

abstractions incorporate general temporal abstraction primitives for detecting the following patterns:

- *Frequency*: more than or fewer than a specified number of values within a given time span.
- *Complex delta-check*: two numerical values whose absolute or percent difference is more than a specified value and/or less than a specific value. The two values may from the same test or two comparable tests, and they must occur within a specified time span.

PROTEMPA took 7 minutes to process a 28-day data set of bedside glucose results and central laboratory glucose results. This data set included 10,782 laboratory glucose results on

5,219 patients. It identified 319 patients with 613 intervals. This amounts to an average of 11 new patients a day. Summary statistics are shown in Table 15. Manual inspection of the results by the author found that all were consistent with the abstraction definitions and the ADA guidelines.

These preliminary results illustrate how PROTEMPA supports detection of a broad range of temporal and mathematical patterns within data sequences, including the detection of patterns in the results of multiple tests. The *complex delta check* temporal abstraction primitive that was implemented for this study (described above) is not one of the standard set of primitives provided by existing temporal abstraction systems (see Section 2.1.2), and is needed for detecting excessive variability between bedside and central laboratory results.

In cooperation with the bedside glucose QA manager at the University of Pittsburgh, a user interface based upon TPOD has been constructed for displaying patients with these intervals, called PatientPatterns (Figure 31). When a user logs in, the display server (Figure 20) retrieves the set of patients with abstractions from the process detector (Figure 20), and returns a web page with a list of the abstractions found and the names and ids of the patients in which they were found. When a patient is selected from the list, the data server retrieves the set of defined data clusters (Section 5.2) from the knowledge base (Figure 20), and for the data clusters whose abstractions have been identified, supporting data are retrieved from the local database cache. Graphs are created to visualize the data clusters, and a web page is returned to the client that displays data clusters in rank order.

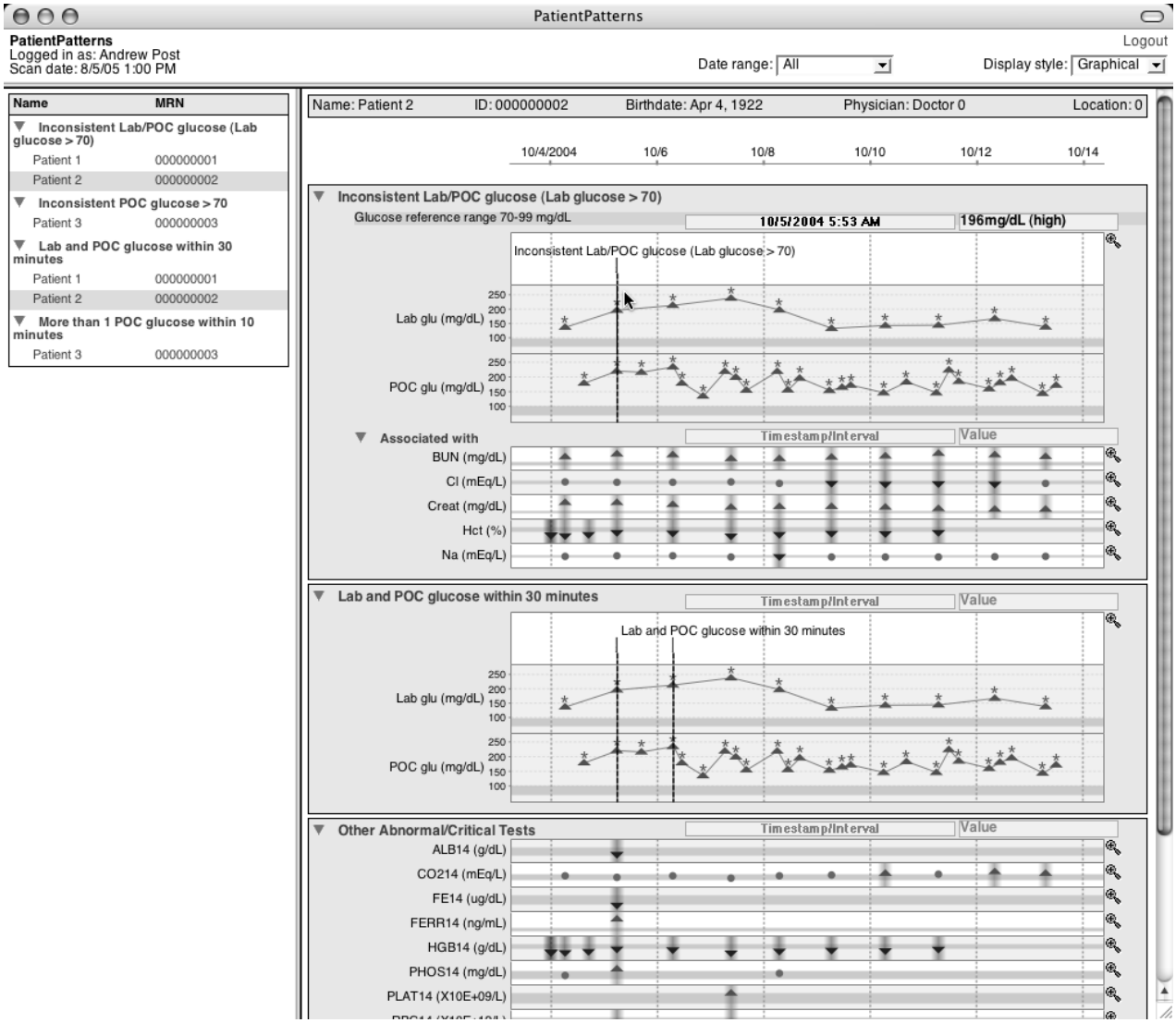


Figure 31: Screenshot of the PatientPatterns user interface. Data is de-identified in this figure, but would be fully identified in a production setting. On the left, four clusters in three sample patients are shown. Selecting a patient shows the clusters and laboratory data for that patient on the right. In the selected patient, there are two instances of contemporaneous laboratory and bedside glucose results (shown in the *Lab and POC glucose within 30 minutes* cluster), and one of them is inconsistent (shown in the *Inconsistent Lab/POC glucose (Lab glucose > 70)*) cluster. Underneath the two clusters for this patient, the *Other Abnormal/Critical Tests* and *Normal Tests* sections show the rest of the patient's data (*Normal Tests* not visible). At the top of the screen, a drop-down box labeled *Date range* sets how far back in time data is shown (1 day, 1 week, etc.). Another drop-down box labeled *Display style* allows switching between the default "Graphical" display based on TPOD, and a standard tabular display of laboratory data (similar to Figure 26). POC = Point of care (bedside).

In PatientPatterns, TPOD functions as a task-specific data display "console" in which only a subset of a patient's data is of primary interest. Development of clinical data displays that support non-patient care activities is a relatively unexplored area of research as compared with research on data displays for results review. As clinical data warehouses become more commonly available as sources of data for clinical research and medical process improvement studies (see Section 4.1), user interfaces to these systems will be needed for querying, retrieving, and analyzing large populations of patients.

PatientPatterns is a prototype. A complete system would include support for automated uploading of bedside test results into the local database cache. It would have the ability to create summaries of entire patient sets for reporting purposes. Optimally, a user interface would be implemented for laboratory personnel to define data clusters for particular QA activities. In this preliminary study, it correctly identified a population of patients with patterns in time sequences of bedside glucose test results that indicate patient care process problems, and visualized the patterns found in each patient. These results are promising, but additional development is needed to create a truly useful production quality QA tool.

APPENDIX A

GLOSSARY

Anemia: a lower than normal number of red blood cells.

Aspartate aminotransferase (AST): an enzyme that is normally present in heart and liver cells. Low levels are also present in the blood. When liver or heart is diseased or damaged, additional AST is released into the blood. A laboratory test for AST levels in the blood is used to check for liver damage.

Bedside testing: Bedside or point-of-care testing is the analysis of blood samples at the bedside, instead of sending samples to the central laboratory for processing.

Blood urea nitrogen (BUN): the amount of nitrogen in the blood that comes from the waste product urea. BUN is produced in the liver, and eliminated by the kidneys. If the kidneys are not able to remove urea from the blood normally, BUN level increases. A BUN test is done to estimate kidney function.

Chloride (Cl): An electrolyte in the blood.

Cholestatic liver disease: liver disease resulting in impairment of bile flow.

Creatinine (Cr, or Creat): Creatinine is a breakdown product of creatine, which is an important component of muscle. Creatinine is eliminated by the kidneys. If the kidneys are not able to remove creatinine from the blood normally, creatinine level increases. A creatinine test is done to estimate kidney function.

Data cluster: part of the temporal process-oriented display. A group of data that are displayed only if a specified group of mathematical and temporal patterns is present.

Digoxin: a drug for treatment of various heart conditions, such as arrhythmias and congestive heart failure. Patients on digoxin have periodic blood tests done to monitor the drug's serum concentration.

HELLP Syndrome: HELLP (Hemolytic anemia, Elevated Liver enzymes, and Low Platelets) syndrome is a severe form of pre-eclampsia that is reflected by abnormalities in laboratory test profiles.

Hematocrit (Hct): the percent of whole blood that is composed of red blood cells. Abnormal hematocrit can be an indication of a variety of conditions, including anemia, blood loss, and dehydration.

Hemoglobin (Hgb): The component of red blood cells that carries oxygen from the lungs to body tissues. A low concentration of hemoglobin in the blood is an indicator of anemia.

Hemolytic anemia: anemia caused by excessive red blood cell breakdown.

Interval: A period of time over which a temporal or mathematical pattern exists, with timestamps specifying the period's start and finish.

Lactate dehydrogenase (LDH): an enzyme found in a variety of human cells, especially in the heart, kidney, liver, and muscle. A test for LDH levels in the blood is used to evaluate the presence of tissue damage.

Mathematical pattern: the specification of a time sequence of clinical data satisfying a set of constraints defined by a mathematical algorithm.

Mathematical relationship: a relationship between the values of two or more clinical data elements.

Na: see Sodium.

Phenytoin: a drug for treatment of epilepsy. Patients on phenytoin have periodic blood tests done to monitor the drug's serum concentration.

Platelets (PLT): a type of blood cell that plays a key role in blood clotting. Platelet count can be affected by many disorders, including hemolytic anemia such as is found in HELLP syndrome.

Process-oriented approach: an integrated approach to improving the detection and visualization of disease and patient care processes that has two components 1) automated identification of temporal and mathematical patterns, and 2) the use of those patterns to control a data display's form, organization, and sequence.

Quality assurance (QA): the process of ensuring reproducible results that are clinically meaningful.

Quinidine: a drug for treatment of cardiac arrhythmias.

Reference range: the desired range of values for a test. The reference range is a property of a test, but may vary for different patients depending on age, gender, or physiologic status. A result out of this range indicates that follow-up investigation is needed.

Sodium (Na): An electrolyte in the blood.

Temporal abstraction: a method for identifying temporal and mathematical patterns within time sequences of clinical data.

Temporal abstraction primitive: a mathematical processing algorithm that specifies constraints on the values of a time sequence of clinical data, potentially with parameters that configure the algorithm for use with specific data types.

Temporal pattern: the specification of a group of intervals with minimum and maximum temporal distances between their endpoints.

Temporal relationship: a relationship between the timestamps of two or more clinical data elements.

APPENDIX B

CASE GUIDELINES

Clinical Case Development Guidelines

These guidelines are written for the expert physicians group collaborating in the research project, "Clinical decision-making using a data-driven display." The project is supported by the National Library of Medicine and runs through September, 2006.

Primary investigator contact information:
James H. Harrison, Jr., MD, PhD
Associate Professor, Pathology Informatics, Department of Pathology
University of Pittsburgh
5230 Centre Ave., 3rd Fl. UPMC Cancer Pavilion
Pittsburgh, PA 15232
office: 412-647-5529 | fax: 412-647-5380 | cell: 412-445-0836
jhrsn@pitt.edu

Project background

We contend that current clinical information system displays are not optimal for working with large amounts of time-series data, such as is typically found in patient records. There is a body of work from outside medicine that suggest specific strategies for optimizing these types of displays, but the applicability of this information in medical environments has not been critically evaluated. We have a long-term goal of improving medical displays by critically evaluating the strengths and weaknesses of current displays and testing the effects of new display strategies on medical decision-making, using standard techniques from human-computer interactions (HCI) studies.

Current study

In this initial study, we will evaluate strategies for improving the display of "valued" time-series data, focusing primarily on laboratory results. This data will be presented in the context of a patient case that includes a history and physical exam, lab results, consultant reports, progress notes and orders. The cases are real, de-identified patient data extracted from UPMC's medical archival system (MARS). The case information will be presented to clinician-subjects using several computer interfaces that differ in key characteristics. The clinicians will be audio-taped and the computer screens will be captured as they review the data using a "think-aloud" style. The recorded events will be coded using a standard HCI approach. At the end of the case review, clinicians will write a set of orders appropriate for the status of the patient at that point in the case. Evaluation will include the effect of the clinical interfaces on both the approach to the case indicated in tapes and screen captures, and the accuracy and completeness of the orders.

Task of the Expert Physician Group

The case data extracted from MARS includes complete cases consisting of H&P, labs, orders, consultant reports, progress notes and discharge summaries. To provide an adequate amount of data to test the interfaces and to make the cases comparable, we recommend that the clinical courses presented to the subjects should be 6-11 days long, with an optimal duration of about 7-9 days. The MARS cases are generally 10-20 days in length, so that a reasonable order-writing point in the middle of the case can be identified. The primary task of the physicians group is to identify this point and validate the orders that should be written then. Although some cases are long and complex, events occurring after the identified order-writing point will not be shown to subjects and are not directly pertinent to this study. The design of the study requires a total of 15 cases, with an initial phase requiring 10 cases. We have 5 members in the physicians group and thus each physician will receive two cases immediately and one additional case as that data becomes available. In addition to creating the three cases, each physician will be asked to review the orders for a total of three cases created by other physicians (so that two physicians see each order set).

Case files

Each case will be supplied as a single plain text file. This file can be opened with Notepad, WordPad, MS Word, or any other word processor or text editor. A text editor that supports line numbering (such as the freeware SciTE <http://www.scintilla.org/SciTE.html> [Windows and Linux/Unix] or TextWrangler <http://www.barebones.com/products/textwrangler/index.shtml> [Mac]) may be useful in discussing the cases. Files will be supplied with Windows/DOS line endings, but Unix or Mac line endings are available on request. The data in the files are arranged chronologically by day and within each day in the sequence: reports/labs/micro/orders. These files are built automatically from multiple MARS output files using a script. Occasionally, you may find typographical errors, duplicate entries or other odd elements. Proofreading is not a task for the physicians group, but if you do encounter a problem, please make a note of it and report it to Jim Harrison (the PI, contact info above).

There is a header at the top of each file, but don't pay too much attention to it because it's mostly for our use and may be confusing. It displays the de-identified "identifier" that is meaningless except that it connects a particular patient's data within the several files that we receive. The end date is the date of the last entry we receive. This may be a discharge summary that's dictated after the patient was discharged. The actual d/c date in the summary, or the date of the last order set, is more reliable. The diagnosis listed in the header is the name of the ICD-9 code that we searched to find the case. It may not be the patient's major diagnosis (the discharge summary is a better source for this).

In each file, there should be a history and physical note early in admission and a discharge summary at the end of the case. These, consultant notes, progress notes and microbiology reports are set off before and after with asterisks and a title indicating the type of report with its date and time. Reports without a specified time are shown as 12:00. Labs and orders are shown as single line entries starting with "Lab:" and "Ord:," respectively. Both start and stop orders are shown. Within the file, the days are shown with a line of equal signs ("=") containing the date.

There is current an overlap between the one-line lab results and the microbiology reports. Ignore this for the moment. We're planning to present the micro reports as text reports, so they won't appear in the display as standard lab values.

Creating the case

Please review each case and determine a point at which it's reasonable to end the case and ask for orders (ideally after a 7-9 day course). This might be the time of an existing order set that could form the basis for the recommended orders. Because this part of the study is particularly concerned with displaying time

course data, it would be beneficial if cases could be ended at a point where the time course of change in lab values contributes in some way to the recommended order set.

If some case data are confusing or otherwise problematic, we would consider deleting them or changing them to a more appropriate value. We would also consider adding data that would improve a case. Finally, if a useful time course could be created by modifying or adding data, we would consider that. However, our preference is to use the actual data as it stands, so we would only change data if it truly contributed to the case. If a case is particularly problematic and really does not suit the purposes of the study, please let Jim Harrison know. We have several cases in reserve that we can use for substitutes if needed.

What should be returned from each case?

1. The case file name
2. The stop time
3. The recommended order set that should be issued at the stop time
4. A problem list that's correct for the stop time would be helpful
5. The key data elements most important in supporting the recommended orders
6. If present, a time course in one or more data elements that is important to recognize
7. Any typos, apparent omissions, or other problems that were noticed (just what you noticed in working with the case, don't worry about proofreading)
8. Any changes you recommend in the case data (if needed)

What happens after that?

1. Once the cases are complete, the text files containing the case data should be deleted. Even though these cases have been de-identified, it is good practice (and required by the IRB) that we do not keep the original data.
2. The physicians group will be paid as consultants. We've discussed general amounts that are consistent with the project's funding, and this can be finalized with Jim Harrison.
3. I think it might be nice to get together for a presentation later in the summer. This could occur at UPMC or we could set up something in a local restaurant in the evening if the group would prefer that. This would give us a chance to go over the study and our preliminary results with you in detail.
4. Once this preliminary phase is complete, I'd like to provide opportunities for those interested to continue to collaborate.

Please feel free to call or email questions or comments to Jim Harrison.

APPENDIX C

IRB CONSENT FORMS

Approval Date: 08/26/04
Renewal Date: 08/25/05
University of Pittsburgh
Institutional Review Board
IRB Number: 0407099

CONSENT TO ACT AS A SUBJECT IN A RESEARCH STUDY

TITLE: **Clinical Decision-making Using a Data-driven Display**

PRINCIPAL INVESTIGATOR: James H. Harrison, Jr., MD, PhD, Assoc. Professor of Pathology
UPMC Cancer Pavilion, #310
5230 Centre Avenue, Pittsburgh, PA 15213; Phone: 412.647.5529
e-mail: harrisonjh@upmc.edu

CO-INVESTIGATORS: Valerie Monaco, PhD, MHCI, Asst. Professor of Medicine
UPMC Cancer Pavilion, #303
5230 Centre Avenue, Pittsburgh, PA 15213; Phone: 413.647.3064
e-mail: monacov@upmc.edu

SOURCE OF SUPPORT: National Library of Medicine

Why is this study being done?

The purpose of this study is to better understand the ways that physicians use computers to care for patients and the characteristics of computer displays that help physicians make correct decisions.

Who is being asked to take part in this study?

UPMC Internal Medicine and Family Practice residents beyond the internship year are invited to participate in this study.

What are the procedures of this study?

If you agree to participate in this research study, you will receive a brief orientation and then you will be asked to complete 6 simulated clinical cases using a computer. The cases will be presented using several different types of computer display and will include a clinical history, presenting symptoms, laboratory tests and results, and a one-week hospital course. You will be asked to analyze the cases using the

computer display and speak your thoughts aloud as you do so. When you have analyzed each case, you should enter a set of clinical orders that best meet the simulated patient's needs. Each case will have a time limit of 15 minutes by which all orders must be entered. To help us more accurately evaluate your use of the computer, we will video/audio tape your actions and verbal responses during each case. When you have completed the cases, there will be a brief "exit interview" in which you will be asked questions about the usability of the computer displays. The testing session should require about two hours to complete.

How will my eligibility for the study be determined?
Medicine or Family Practice residents who are beyond the internship year in UPMC residency programs are eligible to take part in this study.

What are the possible risks and discomforts of this study?

There is little risk involved in this study. No invasive procedures or medications are included. The major potential risk is a breach of confidentiality, but we will do everything possible to protect your privacy. To reduce the likelihood of a breach of confidentiality, all researchers have been thoroughly trained to maintain your privacy. If you are uncomfortable continuing at any point in a session, you are free to stop participating.

Will I benefit from taking part in this study?
You will receive no direct benefit from participating in this study. However, you may learn more about alternative computer displays and the way that you interact with them if you complete the study.

Are there any costs to me if I participate in this study?
There are no costs to you for participating in this study.

How much will I be paid if I complete this study?
If you complete the case simulation session, you will receive a gift certificate from Amazon.com for $100. In addition, the subject with the most accurate clinical orders across all cases will receive a second gift certificate for $100. In the event of a tie in order accuracy, the second gift certificate will go to the subject completing the cases in the least time.

Will anyone know that I am taking part in this study?
All records pertaining to your involvement in this study are kept strictly confidential (private) and any data that includes your identity will be stored in locked files at all times. A number will be assigned to your information and your name will be separated from this coded information during storage. At the end of this study, any records that personally identify you will remain stored in locked files and will be kept for a minimum of five years. Your identity will not be revealed in any description or publications of this research. Although we will video/audio tape the case simulation sessions, the camera will be oriented so as to show the computer screen and not to show your face. We will not refer to you by name during the taping. Most tapes will be transcribed and then destroyed. Some sections of tapes may be preserved for educational purposes if they display important features of physician-computer interaction particularly well.

In unusual cases, your research records may be released in response to an order from a court of law. It is also possible that authorized representatives from the University of Pittsburgh Research Conduct and Compliance Office, the University of Pittsburgh IRB, or the sponsors of this research study (National Library of Medicine) may review your data for the purpose of monitoring the conduct of this study. Also,

if the investigators learn that you or someone with whom you are involved is in serious danger of potential harm, they will need to inform the appropriate agencies, as required by Pennsylvania law.

Is my participation in this study voluntary?
Yes! Your participation in this study is completely voluntary. You may refuse to take part in it, or you may stop participating at any time, even after signing this form. Your decision will not affect your relationship with the University of Pittsburgh or the University of Pittsburgh Medical Center, nor will you lose any benefits that you might be eligible for because of your decision. You may be withdrawn from the study at any time by the investigators: for example, if you were subsequently found to meet any of the study criteria that would exclude you from participating.

How can I get more information about this study?
If you have any further questions about this research study, you may contact the investigators listed at the beginning of this consent form. If you have any questions about your rights as a research subject, please contact the Human Subjects Protection Advocate at the University of Pittsburgh IRB Office, 1-866-212-2668.

****

SUBJECT'S CERTIFICATION

- I have read the consent form for this study and any questions I had, including explanation of all terminology, have been answered to my satisfaction.
- I understand that I am encouraged to ask questions about any aspect of this research study during the course of this study, and that those questions will be answered by the researchers listed on the first page of this form.
- I understand that sessions will be video/audio taped.
  I agree ____ I do not agree ____ to the taping
- I understand that some portions of these tapes may be preserved for educational purposes
  I agree ____ I do not agree ____ to possible preservation of portions of the tapes
- I understand the researchers are often seeking subjects for other studies.
  I agree ____ I do not agree ____ to allow these researchers to contact me about the possibility of participating in other research projects
- I understand that my participation in this study is voluntary and that I am free to refuse to participate or to withdraw my consent and discontinue my participation in this study at any time without affecting my future care at this institution.
- I agree to participate in this study.

_____ _____

Participant's Signature Date

CERTIFICATION of INFORMED CONSENT

I certify that I have explained the nature and purpose of this research study to the above-named individual(s), and I have discussed the potential benefits and possible risks of study participation. Any questions the individual(s) have about this study have been answered, and we will always be available to address future questions as they arise.

_____ _____

Printed Name of Person Obtaining Consent | Role in Research Study

_____ | _____

Signature of Person Obtaining Consent | Date

As part of the research project we will make a digital movie recording of **the computer screen** and **your voice** while you participate in the experiment. Your face does not appear in these digital movies. This recording will be studied by the research team for use in the research project. We would also like you to indicate below what other uses of these digital movies you are willing to consent to. In each of the uses listed below, portions of the digital movie will be used for the purpose of describing the research procedures, and in discussion of research findings. This is completely up to you and your response will in no way affect your payment for participating. We will only use the movies in ways that you agree to. In any use of these movies, your name would not be identified; however, such use does present a risk for loss of confidentiality.

1. The digital movies can be shown at meetings of scientists. _____
   initials

2. The digital movies can be shown in classrooms to students. _____
   initials

3. The digital movies can be shown in public presentations to nonscientific groups. _____
   initials

4. The digital movies can be used on television and radio. _____
   initials

5. The digital movies can be used on a public website maintained by the research group. _____
   initials

I have read the above description and give my consent for the use of the digital movies as indicated above**.**

_____ _____
Participant's Signature Date

_____ _____
Witness Signature Date

APPENDIX D

TRIAL INTRODUCTORY SCRIPT

Open Camtasia and run through setup.

Introduce self.

"My name is <name>. I'm a <title> in the Centers for Oncology and Pathology Informatics, and I'm testing alternative computer displays that present clinical information."

Give general overview.

"These computer displays have been developed to test how physicians use computers in making patient care decisions. I'm going to ask you to use two different displays to review and write orders on simulated patients. The purpose of the study is to determine how the differences in the displays affect the way you analyze the cases and make decisions about the orders that need to be written for these patients."

"I am testing the computer displays. I am not testing you. I'm looking for places where the displays might be difficult to use or might influence your approach to a case, so if you can't do some things, please don't feel bad. That sort of thing is exactly what we are looking for."

"In this observation, I am interested in what you think about as you do the task. I'm going to ask that you 'think aloud' while you are using the display. This means that you will say what you are thinking out loud while you analyze each case and write orders. I'm going to give you more information about thinking aloud in a few minutes."

"During the session, we will be making a video recording of what is happening on the computer screen. We will also be recording your voice."

"Remember, this is completely voluntary, so if you feel uncomfortable at any time, feel free to stop. Also, your identity will be kept confidential unless you indicate on the consent form that we may present the digital recordings from this session in different settings. Presenting the recordings may present a risk for loss of confidentiality."

"In addition to the case simulations, we will ask you to participate in a brief exit interview."

"The entire session should take about 2 hours to complete."

"You will be provided with a $100 gift certificate if you complete the session."

Give consent form.

"Please read over this consent form. It tells you in more detail about what will happen at today's session. If you sign it, it means you understand the purpose of the study and how the data will be used. You will receive a copy of this form with contact information, if you wish to discuss or report any issues that arise from your participation in the study."

"The last page of the consent form asks for permission to use the digital movie in different settings. This is completely up to you and your response will in no way affect your payment for participating. We will only use the movies in ways that you agree to."

If consent is signed, then…

Deliver think-aloud instructions.

"What I mean by 'think aloud' is that I want you to tell me **everything** that you are thinking from the first time you view each case display until you finish. I would like you to speak constantly from the time I give you the task until you have completed it. I don't want you to try to plan what you say, or try to explain to me what you are saying. Just act as if you are alone, speaking to yourself…just a little louder."

Demonstrate thinking aloud.

"Let me demonstrate thinking-aloud for you as I try to multiply 42 x 22 in my head."
<demonstrate>

"Now you try thinking aloud: Here's a problem: please think aloud while you answer the question, How many windows are there in the place where you live?"

"Now, those problems were solved entirely in our heads. However, when you are working on the computer, you'll also be looking for things and seeing things that catch your attention. These things you are searching for and things that you see are as important for our observation as thoughts you are thinking from memory, so please verbalize these too. For example, listen to the types of things I say as I think-aloud when I disable the screen saver on this computer."
<demonstrate>

"Do you have any questions about thinking aloud?"

Case simulation instructions. Login to the system using the subject's username and password. Ask subject to sit down in front of computer.

“Imagine the following scenario. It is June 1, and you are covering six patients for another colleague. You are rounding on them for the first time. For each case, you have 15 minutes to determine the current status of the patient and write a set of orders for what to do next. All you have to work with is the information presented to you in the simulated case. Unlike a real case, you cannot take your own history or do your own physical exam.”

“At the beginning of the session, the computer will display a ‘Home’ window prompting you to begin the session. The first two cases are for orientation and practice, the last four are the test cases. You must complete the cases in order. After you complete each case, you will return to the Home window to start the next case.”

“As you do the practice cases, I will introduce you to the interface, and I will be able to answer any questions that you have about the operation of the display. However, as you do the test cases, I won’t be able to answer any further questions. But if you do have questions, go ahead and ask them anyway so I can learn more about what questions are prompted by the computer displays. I’ll answer your questions after we’re finished. Also, if you forget to think aloud, I’ll say, ‘Please keep talking.’”

“You have 15 minutes to review each test case and write orders for it. I will announce the time remaining two minutes and one minute before the end of each case. After 15 minutes, the computer will return you to the Home window.”

Open first practice case (spend no more than 15 minutes on it). Check time on watch.

“The case display screen shows patient demographic information at top in the center. The time you have left is displayed to the left of the demographics. This field will turn red when you have one minute left. The admit date and current date for the case are also displayed near the top. There are tabs for ‘Summary’, ‘H & P, ‘Progress notes.’ ‘Labs’, ‘Micro’, ‘Current Rx’, ‘Previous Orders’, ‘Radiology,’ ‘EKG,’ and ‘Other Documents.’ The ‘Summary’ tab is the first tab displayed by default. It contains a 1-2 sentence summary of the case and a problem list. ‘Other Documents’ contains consultation reports, surgical pathology, vascular lab, etc. All information except the laboratory studies is presented as text. <demonstrate>”. At the top-left of the screen is a hyperlink called ‘Orders’, which you click to get to the order-writing page. <demonstrate> Click the button ‘Return to case’ to return to the case. <demonstrate>”

“The ‘Progress Notes’, ‘Micro’, ‘Radiology’, ‘EKG’, and ‘Other Documents’ tabs all contain documents. A document list on the left side of the tab shows the available documents in reverse chronological order (most recent document at the top, and the current document highlighted in yellow). Note that some progress notes may be missing from the case. The ‘Current Rx’ tab shows orders that are currently valid. The ‘Previous Orders’ tab shows one-time orders and orders that have been stopped. Both show orders in reverse chronological order (most recent orders at the top). Note that we were unable to obtain blood bank order and administration data for this study, so they are not present in the display.”

“In some cases, laboratory results will be presented using a combination of graphs and timelines. <demonstrate> All results in this display are aligned to a time axis at the top of the screen that is

in forward chronological order (most recent results are on the RIGHT side of the graph). At the top of the laboratory display, some results will be organized according to problem patterns found in the data by our software. Patterns are shown according to a ranking chosen by our software (most important pattern first). The first pattern is fully revealed. Subsequent patterns are collapsed; clicking on the name of the pattern will reveal it. Timelines show the temporal extent of patterns and delimit the raw data values in which they were found. Raw data values are shown in standard XY plots (time on the x-axis). 'Associated data' show other data elements that may be useful in interpreting the pattern. Underneath these patterns, in the 'Labs by Group' section, you will find labs organized similarly to production electronic medical records. Clicking on the name of a group will reveal it."

"In the graphical display, upward-pointing red triangles signify high values. Downward-pointing blue triangles signify low values. Dark-gray circles signify normal values. A blue background in a graph signifies the normal range. The magnifying glass next to each graph may be clicked to vertically expand the graph to see more detail. Running the cursor along the graph will cause the timestamps and numerical values for each point to be displayed in boxes at the top of the current group. You can tell which value is currently 'selected' by the presence of a vertical black line. Clicking on a value will reveal a popup box with more details about the value (numerical value, timestamp, normal range, etc.)."

"Graphs of associated data and all of the graphs in the 'Labs by Group' section are initially shown in compressed form. No line is drawn between the points, and there is additional red or blue shading behind abnormal values. To expand these to full XY plots, click the magnifying glass icon."

"When you are ready to write an order, click the 'Orders' button. A text entry box will appear in which you should type your orders. Two buttons below the text box, 'Submit and go to next case' and 'Return to case', will bring you to the next case and return you to the current case, respectively. Orders may be entered or deleted at any time prior to completing the case by editing the text in this window. Orders will accumulate until the case is finished, so you may enter orders as you go, or all at the end, as you desire."

"When you feel your orders are complete, click "Submit and go to next case" and the orders will be stored in the system for that case. You will then return to the 'Home' window and may start the next case."

If there is time remaining, subject may use the remaining time to explore the interface.

Advance to the practice tabular case (spend no more than 15 minutes on it). Check time.

"In other cases, laboratory studies may be presented in a spreadsheet-like display with columns of results organized by date and group. The type of display used for a particular case is chosen by the system. Your task is to use these displays to understand the simulated patient and then write orders appropriate for that patient. Be sure that you speak your thoughts aloud as you use the display. In the tabular display, values are grouped as in production laboratory results software. Groups may be revealed or hidden by clicking on the header text <demonstrate>.

Results are displayed in reverse chronological order (most recent results are on the LEFT side of the table). Abnormal values are bold-faced with an ‘H’ for high and a ‘L’ for low. The patterns revealed in the graphical display are not visible in the tabular display.”

Subject may use the remaining time to explore the interface and write orders for this case. Be sure to warn at 2-minute and 1-minute mark.

“After you complete all the cases in the session, I will interview you briefly to record your reactions to the cases and the computer displays.”

“As an incentive to work quickly but accurately, the resident with the greatest accuracy of orders in the least time will receive an additional $100 gift certificate.”

“A final word before we begin: some of the cases are complex and may not be able to be finished completely in 15 minutes. This is intended and will help to highlight the differences in the two laboratory display styles. You should just try to do your best, verbalize your thoughts, try not to get bogged down, and keep in mind that the study is not meant to evaluate you personally.

“Do you have any questions about the case simulations?”

Begin recording. Make sure to watch screen for the 2-minute and 1-minute marks.

“You may begin now.”

In between cases, tell the subject, “Please proceed to the next case when you are ready.”

APPENDIX E

TRIAL EXIT SCRIPT

Thank you for participating in our study. As you know the primary purpose of these sessions is to record your detailed thoughts and responses as you review simulated cases using the computer displays. We are also interested in any general reactions you have to the cases and displays that might not be clearly captured in the think-aloud session. For example:

What is your overall reaction to the clinical displays?

Which display would you rather work with on a regular basis?

Have you worked with electronic clinical displays previously? Which ones?

What is your reaction to the simulated cases?

Do you believe the simulated cases provided a reasonable test of the displays? If not, why not?

Are there features in these displays that you particularly liked? Which ones and why?

Are there features in the displays that you disliked? Which ones and why?

Are there any other comments you would wish to make about the study?

Stop recording (press F10).

APPENDIX F

CODED "THINK-ALOUD" TRANSCRIPT

The following is a transcript of a subject reviewing case 3 (Table 5). Codes are in square brackets. Numbers in angle brackets followed by ¤ are timestamps in seconds since the start of the case. Two-minute and one-minute time warnings are printed in italics.

[BR][RE]75-year-old with [RC]cholangitis, [RC]pneumonia, [RC]diabetes and [RC]hyperglycemia. [RC]Morbid obesity, [RC]type II diabetes, [RC]hypertension, [RC]Afib, [RC]ascending cholangitis, [RC]right upper lobe pneumonia, [RC]deconditioning, [RC]stage II decubitus ulcer and he's [VA]75. [RE]The 23rd and this is the [IC-DI]8th today.[CBR]

¤<17116>[GS-H][BR]He presented [SE]it doesn't say the chief complaint -- [RE]by way of paramedics because of [RE]upper abdominal discomfort and [RE]fullness. [RE]Tried to belch, [RE]could not pass it. [RE]Began substernal chest discomfort, [RE]shortness of breath, [RE]no diaphoresis.

Okay, [RE]these are our history which [VA]I already reviewed -- [RE]positive for breast lumpectomy, [RC]family history positive for myocardial infarction. [RE]Medications: [RE]cardizem, [RE]Zocor, [RE]neurontin, [RE]axid, [RE]hydralazine, [RE]lexapro and [RE]enteric-coated aspirin. [RE]Negative for jaundice, [RE]night sweats, [RE]bruising, [RE]pyrosis, [RE]odynophasia, [RE]no swollen ..., _____ -- [IC-MV]all was negative.

[RE]Vital signs: blood pressure [RE]113, [IC-STM]she's stable hemodynamically. [RE]Fundi's... there is no... there doesn't appear to be . . . [RE]they don't mention any jaundice here, [RE]basilar crackles, [RE]no wheezing. [RE]S1 [RE]S2 present, [IC-MV]cardiac is normal, [RE]abdomen is obese, [RE]no dominant mass, [RE]negative HSM. [RE]Tenderness in epigastrium and [RE]right upper quadrant, [RE]both without rebound. [IC-RV]The breast is normal. [IC-MV]Neuro exam is normal. [RE]Pelvic deferred. [IC-MV]Significant edema [RE]2+ extending [RE]1 + to upper calves bilaterally. [RE]Both feet, okay. [RC]Chronic venous stasis, [RE]changes of integument over the skin, [RC]which is a chronic problem for this patient. [RRC]Chronic venous stasis, [RRC]changes of the integument, okay.

Found to have [RE]on CT chest to [RC]right upper lobe infiltrate, [RC]pneumonia, [RC]bilateral atelectasis. [RC]Start treatment azithro and [RC]cefurox. [RE]Did have nitrous sprays which did [RE]cause her to have some relief of substernal discomfort. [RE]White cell count was 64,000 or 6.4 thousand I can't make that out. [RE]74 polys, [RE]1 eosinophil, [RE]H&H of 37.8 and [RE]12.8 that's reversed. Platelets are [RE]177. [RE]Troponin 0.1 and a [RE]total bili of 3.3.

[RE]PLT 210, [RE]AST, [RE]alkaline phosphatase is 289 and [IC-ST]GGTP is high, I think likely.

[RE]She was admitted, [RE]monitored, [RC]evaluation of chest pain syndrome, [RE]troponin serially, [RE]pulse oximetry, [RC]she has pneumonia, [RC]placed on antibiotics, [RC-STM]elevated liver function, [RC]serial LFT, [RC]diabetes mellitus, okay. So [CBR]

¤<165403>[GS-PN1][BR][RE]problem list

¤<167206>[GS-PN11][RE]So this was her admission and at that time, let me go straightaway to the evaluation. She was on _____ [RE]ERCP. There was [RE]stone, [RE]sludge and [RE]pus as well, so [EDA]she did have ascending cholangitis that was definitely there.

¤<182219>[GS-PN10][RE]Recent episode of ERCP [IC-MV]marked improvement. She is [RE]sitting up, [RE]joking with the hospital staff [EDA]that's excellent. So she was [RC]started on Zosyn, [RC]heparin, [RC]propanolol, [RC]protonix, okay. [RRC]Ascending cholangitis, [VA]ERCP, [VA]she was doing better. Prior to going to the operating room, [RE]request that she be given cardiac clearance.

¤<206869>[GS-PN9]So back to her service [RC]acute cholangitis [RE]pending clearance surgical intervention. [RC]Recommended cardiac catheterization, [RC-STM]her vital signs are stable let us seen. Patient does have [RC]positive MI and is [RC]for cardiac catheritization. Has [RC]cholangitis too, has [RC]diabetes mellitus, has [RC]hypertension [RC]thought to be stabilized on present regime.[CBR]

¤<232667>[GS-PN1][SE]Let us see what she is doing now. [RE]The GI service is currently following [RC]ascending cholangitis and [RC]choledocholithiasis, who underwent [RC]successful [RE]ERCP and [RE]stone extraction. [RE]Resting comfortable and [RE]easily arousable. [IC-ST]Temperature is fine, [IC-ST]pulse good, [IC-ST]respiration good, [IC-STM]blood pressure good. [RE]Sodium or [IC-STM]electrolyte here as per record look good, [IC-CN]no positive cultures. [VA]She's on the same medication I saw earlier. [IC-MV]Examination appears normal. [RC]She has venous stasis also, which is . . . and [RE]she underwent on 31st her cardiac catheterization. [RC]No significant disease.[CSE]

[BR][RC]History of ascending but has [RE]ecently undergone left heart catheterization which was [RC]negative.... [RE]Followup outpatient evaluation scheduled in either a [RE]laparascopic or [RE]open chole. During that time she will undergo [RC]preoperative testing to prepare her for that. We'll sign off for now. [RE]Please reconsult if needed.[CBR]

¤<291835>[GS-L][BR]So let us quickly review her labs okay. It's taking time. [CBR][BR]Let me in fact

¤<302319>[GS-C]review her current therapies. So she's on [RE]Lasix, [RE]she's on diphenhydramine, [RE]diazepam [QD]why is she on diazepam 5mg? [RE]Glipizide, [RE]lisinopril, [RE]pantoprazole, [RE]aspirin, [RE]propranolol, [RE]simvastatin, [RE]heparin she's q8 hr. [RE]Dextrose. She's on [RE]Zosyn and she's on [RE]sodium chloride. [CBR]

¤<326175>[GS-PO][BR][RE]Previous orders _____. There's too much here to review for me. [CBR]

¤<332758>[GS-R1][BR]

¤<335340>[GS-R4][IC-MV]So her initial chest x-ray was normal.

¤<338390>[GS-R3]Her [RE]CT abdomen did show that she had [RC]right upper lobe pneumonia and [RC]mild dilitation of... [RC]pulmonary artery hypertension and [RC]cholelithiasis.

¤<347879>[GS-R2][RE]Ultrasound showed [RC]cholelithiasis [RC]without evidence of cholecystitis and that was [RE]done on the 24th.

¤<353340>[GS-R1]And just on [RE]1st showed [RC]mild edema, findings compatible with [RC]CHF and [RC]cardiac enlargement. Okay, [CBR]

¤<360826>[GS-E1][BR][RE]EKG [RC]normal sinus, [RC]right bundle branch block, [RC]left anterior fascicular block, [RC]T wave, okay. [CBR]

¤<367941>[GS-O1][BR]Is there any other document? [RE]Cardiac catheterization. She had [RC]no significant epicardial CAD, [RC]normal LV function, [IC-ST]troponin elevation. [RE]That's all Cardiology has written.[CBR]

¤<382399>[GS-E1][RV][CRV]

¤<383714>[GS-R1][RV]I've already reviewed this. [CRV]

¤<386009>[GS-PO][RV]I've already reviewed this. [CRV]

¤<387335>[GS-C][RV][RE]Her current therapy is this. [CRV][SE]Let us see if she is on aspirin. [RE]Okay. [CSE]

¤<393867>[GS-M][RV][CRV]

¤<395325>[GS-L][RV][CRV]

¤<397127>[GS-PN1][SE]Let me see the progress right now, what is she having? [RE]So she's comfortable and [RE]easily arousable. [CSE]

¤<406736>[GS-PN2][QD]Why is that? [SE]Let us see what happened to her on the day earlier. [ISE][BR]So she has [RC]stage II decubitus ulcer in the buttocks, okay, she will need a consult for... [RE]cardiac cath scheduled today for [RE]clearance, [RE]GI surgery which will be [RE]done after holidays.

So [RE]Zosyn is to 8. [RC]Change to levaquin po d/c, [RE]follow up January. [RC]Possible UTI. [RE]Urine cloudy in foley, [IC-ST]patient afebrile. [RE]Will obtain . . . so they did that. [RC]Possible C. diff. [QD]Why are they thinking C. diff? [RC]Deconditioning. [RC]Will order

home care. [RE]PT/OT to eval, _____ needs change and _____. [RC]Plan is to discharge patient home tomorrow. [EMA]So the day before yesterday,

¤<455386>[GS-PN1]the plan was to discharge. [CBR][SE]And what is the plan today? Plan today [RE]follow up as an outpatient. During that time she will undergo preoperative testing, will sign off for now. [CSE][RV]So I believe that she's doing well.

¤<477972>[GS-C]So her current regimen, I will change her orders . . . let us see. [VA]She's eating well. And we can change her orders and [EMA]we can discharge this patient home. [CRV]

¤<493026>[GS-NO]Okay, let's see.

¤<496715>[GS-S]

¤<498264>[GS-C][RV]First of all, [EDA]she did have some pulmonary edema [EDA]but she has CHF too.[CRV][MD] [EMA]I'll give her 40 mg b.i.d. just po. So

¤<511889>[GS-NO][RE]orders [EMA]d/c IV, change, [EMA]start 40 mg po b.i.d. [CMD]

¤<537397>[GS-S]

¤<539644>[GS-C][RV]What she needs. [RE]Diphenhydramine [EMA]I don't have any problem.[CRV][MD] [EMA]I'll d/c the

¤<543749>[GS-NO]diazepam. [EMA]This doesn't need that. [CMD]

¤<552445>[GS-S]

¤<554479>[GS-C][RV][EMA]And glipizide is fine with me, [RE]lisinopril is 10 mg. [EMA]Pant... Protonix is okay, [EMA]aspirin is fine, though [EMA]she can reduce it at [RE]325 but she's in _____. [RE]Propranolol 20, [RE]simvastatin 20, [RE]heparin [EMA]heparin can be d/c'ed now since she is going out. [RE]Dextrose and [RE]pip/taz. -- [EMA]we can change the pip/taz to . . . [CRV][SE]what did they say?

¤<591053>[GS-PN1][RRC]They said to change it to levaquin? Was it levaquin? Let me see here.

Where is the . . .[RE]assessment and plan, [RE]outpatient evaluation, [RE]laparascopic chole and

¤<605982>[GS-PN2][RE]it was on 31st, [RRC]I believe they said that they would [RC]change to levaquin po 4 d/c, then [RE]follow up in January. [EDA]Because she's pain free okay. [CSE]

¤<616207>[GS-C]

¤<617230>[GS-NO][MD][EMA]So we will d/c the Zosyn, [EMA]start levaquin 100 mg. She's also [EMA]]give her only 250 po q day and to 7 days.[CMD]

¤<648663>[GS-S][RV]What else she needs?

¤<650516>[GS-C][SE]Let me see her

¤<651709>[GS-S]current medications. Medications on admission were [RE]cardizem, [RE]Zocor, [RE]neurontin, [RE]Axid, [RE]hydralazine, [RE]lexapro, [RE]enteric-coated aspirin.

¤<663930>[GS-C]Right now she's [RE]on the cardizem, where was it? [IC-CN]She's not on any . . . calcium channel blockers.

¤<674650>[GS-H] _____ start her . . . So it was [RE]cardizem... [CRV][MD]I don't know to do this but [EMA]I'll just write for it that she resume all medication. [RE]Zocor, [RE]neurontin, [RE]hydralazine, [RE]lexapro,

¤<686955>[GS-NO][RE]enteric-coated aspirin.[CMD]

¤<688756>[GS-S]Return to case. [RV][SE]Let me see the

¤<690906>[GS-PN1]progress note. [SE]Let me see . . .

¤<695229>[GS-PN2][RE]31st, okay. [RC]Possible UTI, [RC]possible C. diff.[CRV]

¤<703876>[GS-M][SE][QD]But they said C. diff but [IC-CN]I didn't see any microbiology report. [CRV]

¤<706965>[GS-L]Let us see if there is anything here. [QD]Why were they thinking C. diff if there was [RC]no diarrhea. And let us see [ICN-STM]the CBC is not high. I don't think . . . [ICN-ST]WBC is normal. [EDA]I don't think they should think about that. [RE]Differential . . . I don't see anything here. [ICN-STM]It is all normal.

[ICN-STM]Coags is fine. [CSE][BR]___ hepatitis. Was a hepatitis panel _____ done on her? [ICN-MV]The items all negative. Okay, [RE]urinalysis... let's see what her urinalysis that was initially done. Okay. What is this? [RE]White blood cells was 37 but that was [RE]at the time of her admission, I believe, when she was admitted. [EMA]She has been very nicely treated for that. [EMA]So I'm not going to chase that at this time. Three minutes, okay. [QD]Was a U-tox done? I'm not sure. [CBR]Okay, [RV]let us

¤<777807>[GS-PN1]go back to the progress note and

(2 minutes)

Okay, thank you.

¤<784452>[GS-PN3]Just make sure I'm doing everything, so. . . [RE]accuchecks and

¤<790714>[GS-PN2]so what all they said? [RC]Change to levaquin [EDA]I'm not thinking of C. diff at this time. [SC]PT/OT [CRV][MD][EMA]I'll give her home PT/OT, [EMA]follow up with GI clinic. PT/OT follow up with GI clinic and as an outpatient, okay.

¤<815342>[GS-NO][EMA]PT/OT therapy consult. [EMA]Home consult. [EMA]Follow up with GI clinic in one month.

(1 minute)

She also needs . . . what is that . . . [EMA]wound care consult prior to d/c. [EMA]Followup PCP in 14 days. [EMA]D/c to home after seen by appropriate people. And what else should be _____? Let's see. [EMA]Home care consult. Okay.[CMD]

BIBLIOGRAPHY

1. Shahar Y. Dimensions of time in illness: an objective view. Ann Intern Med. 2000 Jan 4;132(1):45-53.

2. Aliferis CF. A temporal representation and reasoning model for medical decision-support systems [Ph.D.]. Pittsburgh, PA: University of Pittsburgh; 1998.

3. Bates DW, Cohen M, Leape LL, Overhage JM, Shabot MM, Sheridan T. Reducing the frequency of errors in medicine using information technology. J Am Med Inform Assoc. 2001 Jul-Aug;8(4):299-308.

4. Nigrin DJ, Kohane IS. Temporal expressiveness in querying a time-stamp--based clinical database. J Am Med Inform Assoc. 2000 Mar-Apr;7(2):152-63.

5. Dorda W, Gall W, Duftschmid G. Clinical data retrieval: 25 years of temporal query management at the University of Vienna Medical School. Methods Inf Med. 2002;41(2):89-97.

6. Safran C, Chute CG. Exploration and exploitation of clinical databases. Int J Biomed Comput. 1995 Apr;39(1):151-6.

7. Murray MD, Smith FE, Fox J, Teal EY, Kesterson JG, Stiffler TA, et al. Structure, functions, and activities of a research support informatics section. J Am Med Inform Assoc. 2003 Jul-Aug;10(4):389-98.

8. Bui AA, Weinger GS, Barretta SJ, Dionisio JD, Kangarloo H. An XML Gateway to Patient Data for Medical Research Applications. Ann N Y Acad Sci. 2002 Dec;980:236-46.

9. van Mulligen EM, Timmers T, van Bemmel JH. User evaluation of an integrated medical workstation for clinical data analysis. Methods Inf Med. 1993 Nov;32(5):365-72.

10. Schubart JR, Einbinder JS. Evaluation of a data warehouse in an academic health sciences center. Int J Med Inf. 2000 Dec;60(3):319-33.

11. Shortliffe EH, Perrault LE. Medical Informatics: Computer Applications in Health Care and Biomedicine. 2nd ed: Springer-Verlag; 2001.

12. Augusto JC. Temporal reasoning for decision support in medicine. Artif Intell Med. 2005 Jan;33(1):1-24.

13. O'Connor MJ, Grosso WE, Tu SW, Musen MA. RASTA: a distributed temporal abstraction system to facilitate knowledge-driven monitoring of clinical databases. Medinfo. 2001:508-12.

14. Haimowitz IJ, Kohane IS. Managing temporal worlds for medical trend diagnosis. Artif Intell Med. 1996 Jul;8(3):299-321.

15. Larizza C, Moglia A, Stefanelli M. M-HTP: a system for monitoring heart transplant patients. Artif Intell Med. 1992;4:111-26.

16. Shahar Y, Musen MA. Knowledge-based temporal abstraction in clinical domains. Artif Intell Med. 1996 Jul;8(3):267-98.

17. Bellazzi R, Larizza C, Magni P, Montani S, Stefanelli M. Intelligent analysis of clinical time series: an application in the diabetes mellitus domain. Artif Intell Med. 2000;20(1):37-57.

18. Kahn MG, Fagan LM, Sheiner LB. Combining physiologic models and symbolic methods to interpret time-varying patient data. Methods Inf Med. 1991 Aug;30(3):167-78.

19. Hayes-Roth B, Washington R, Ash D, Hewett R, Collinot A, Vina A, et al. Guardian: a prototype intelligent agent for intensive-care monitoring. Artif Intell Med. 1992;4:165-85.

20. Carrault G, Cordier MO, Quiniou R, Wang F. Temporal abstraction and inductive logic programming for arrhythmia recognition from electrocardiograms. Artif Intell Med. 2003 Jul;28(3):231-63.

21. Martins SB, Shahar Y, Galperin M, Kaizer H, Goren-Bar D, McNaughton D, et al. Evaluation of KNAVE-II: A Tool for Intelligent Query and Exploration of Patient Data. Medinfo. 2004:648-52.

22. Shahar Y, Cheng C. Model-based visualization of temporal abstractions. Comput Intell. 2000 May;16(2):279-306.

23. Fackler J, Kohane I. Monitor-driven data visualization: SmartDisplay. Proc Annu Symp Comput Appl Med Care. 1994:939-43.

24. Chakravarty S, Shahar Y. Acquisition and analysis of repeating patterns in time-oriented clinical data. Methods Inf Med. 2001;40(5):410-20.

25. Kawasaki S, Ho TB, Nguyen DT. Abstraction of Long-Term Changed Tests in Mining Hepatitis Data. 7th Int Conf on Knowledge-Based Intelligent Information and Engineering Systems (KES 2003); 2003; 2003. p. 366-72.

26. Wyatt JC, Wright P. Design should help use of patients' data. Lancet. 1998 Oct 24;352(9137):1375-8.

27. Connelly DP, Sielaff BH, Willard KE. A clinician's workstation for improving laboratory use. Integrated display of laboratory results. Am J Clin Pathol. 1995 Sep;104(3):243-52.

28. Zeng Q, Cimino JJ, Zou KH. Providing concept-oriented views for clinical data using a knowledge-based system: an evaluation. J Am Med Inform Assoc. 2002 May-Jun;9(3):294-305.

29. Elson RB, Connelly DP. The impact of anticipatory patient data displays on physician decision making: a pilot study. Proc AMIA Annu Fall Symp. 1997:233-7.

30. Elting LS, Martin CG, Cantor SB, Rubenstein EB. Influence of data display formats on physician investigators' decisions to stop clinical trials: prospective trial with repeated measures. BMJ. 1999 Jun 5;318(7197):1527-31.

31. Martin J. The Wrong Use of Automation. The Great Transition. New York: AMACOM; 1995. p. 32-41.

32. Ellis SR, McGreevy MW, Hitchcock RJ. Perspective traffic display format and airline pilot traffic avoidance. Hum Factors. 1987;29(4):371-82.

33. Schwartz DR, Howell WC. Optional Stopping Performance under Graphic and Numeric CRT Formatting. Hum Factors. 1985;27(4):433-44.

34. Bennett KB, Walters B. Configural display design techniques considered at multiple levels of evaluation. Hum Factors. 2001 Fall;43(3):415-34.

35. Kleinmuntz DN, Schkade DA. Information Displays and Decision-Processes. Psychological Science. 1993 Jul;4(4):221-7.

36. Schkade DA, Kleinmuntz DN. Information Displays and Choice Processes: Differential Effects of Organization, Form, and Sequence. Organ Behav Hum Decis Process. 1994;57:319-37.

37. Bergus GR, Chapman GB, Levy BT, Ely JW, Oppliger RA. Clinical diagnosis and the order of information. Med Decis Making. 1998 Oct-Dec;18(4):412-7.

38. Woolf SH. Practice guidelines: a new reality in medicine. I. Recent developments. Arch Intern Med. 1990 Sep;150(9):1811-8.

39. Sibai BM. The HELLP syndrome (hemolysis, elevated liver enzymes, and low platelets): much ado about nothing? Am J Obstet Gynecol. 1990 Feb;162(2):311-6.

40. Haux R. Health information systems - past, present, future. Int J Med Inform. 2006 Mar-Apr;75(3-4):268-81.

41. Sax U, Schmidt S. Integration of genomic data in Electronic Health Records--opportunities and dilemmas. Methods Inf Med. 2005;44(4):546-50.

42. Hall A, Walton G. Information overload within the health care system: a literature review. Health Info Libr J. 2004 Jun;21(2):102-8.

43. Miller RA. Medical diagnostic decision support systems--past, present, and future: a threaded bibliography and brief commentary. J Am Med Inform Assoc. 1994 Jan-Feb;1(1):8-27.

44. Wolfram DA. An appraisal of INTERNIST-I. Artif Intell Med. 1995 Apr;7(2):93-116.

45. Yu VL, Buchanan BG, Shortliffe EH, Wraith SM, Davis R, Scott AC, et al. Evaluating the performance of a computer-based consultant. Comput Programs Biomed. 1979 Jan;9(1):95-102.

46. Hanson CW, 3rd, Marshall BE. Artificial intelligence applications in the intensive care unit. Crit Care Med. 2001 Feb;29(2):427-35.

47. Shahar Y, Miksch S, Johnson P. The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artif Intell Med. 1998 Sep-Oct;14(1-2):29-51.

48. Musen MA, Tu SW, Das AK, Shahar Y. EON: a component-based approach to automation of protocol-directed therapy. J Am Med Inform Assoc. 1996 Nov-Dec;3(6):367-88.

49. Miller RA, Waitman LR, Chen S, Rosenbloom ST. The anatomy of decision support during inpatient care provider order entry (CPOE): empirical observations from a decade of CPOE experience at Vanderbilt. J Biomed Inform. 2005 Dec;38(6):469-85.

50. Harrison JH, Jr., Rainey PM. Identification of patients for pharmacologic review by computer analysis of clinical laboratory drug concentration data. Am J Clin Pathol. 1995 Jun;103(6):710-7.

51. Connelly DP. Embedding expert systems in laboratory information systems. Am J Clin Pathol. 1990 Oct;94(4 Suppl 1):S7-14.

52. Hripcsak G, Clayton PD, Jenders RA, Cimino JJ, Johnson SB. Design of a clinical event monitor. Comput Biomed Res. 1996 Jun;29(3):194-221.

53. Shahar Y. A Knowledge-based Method for Temporal Abstraction of Clinical Data [Doctoral]. Palo Alto, CA: Stanford University; 1994.

54. Aliferis CF, Cooper GF, Miller RA, Buchanan BG, Bankowitz R, Giuse N. A temporal analysis of QMR. J Am Med Inform Assoc. 1996 Jan-Feb;3(1):79-91.

55. Vilain M, Kautz H, van Beek P. Constraint propagation algorithms for temporal reasoning: A revised report. In: Weld DS, de Kleer J, editors. Readings in Qualitative Reasoning about Physical Systems: Morgan Kaufmann; 1989. p. 373-81.

56. Shahar Y. A framework for knowledge-based temporal abstraction. Artif Intell. 1997;90:79-133.

57. Combi C, Pinciroli F, Pozzi G. Managing different time granularities of clinical information by an interval-based temporal data model. Methods Inf Med. 1995 Dec;34(5):458-74.

58. Stillman J, Arthur R, Deitsch A. Tachyon: A constraint-based temporal reasoning model and its implementation. SIGART Bulletin. 1993;4(3):T1-T4.

59. Dechter R, Meiri I, Pearl J. Temporal constraint networks. Artif Intell. 1991;49:61-95.

60. Cormen TH, Leiserson CE, Rivest RL. Introduction to Algorithms. Cambridge, Mass.: MIT Press; 1990.

61. Allen JF. Maintaining Knowledge about Temporal Intervals. Commun ACM. 1983;26(11):832-43.

62. Kuilboer MM, Shahar Y, Wilson DM, Musen MA. Knowledge reuse: temporal-abstraction mechanisms for the assessment of children's growth. Proc Annu Symp Comput Appl Med Care. 1993:449-53.

63. Horn W, Miksch S, Egghart G, Popow C, Paky F. Effective data validation of high-frequency data: time-point, time-interval, and trend-based methods. Comput Biol Med. 1997;5:389-409.

64. Kahn MG, Marrs KA. Creating temporal abstractions in three clinical information systems. Proc Annu Symp Comput Appl Med Care. 1995:392-6.

65. Bellazzi R, Larizza C, Riva A. Temporal Abstractions for Interpreting Diabetic Patients Monitoring Data. Intelligent Data Analysis. 1998;2(1-4):97-122.

66. Boaz D, Balaban M, Shahar Y. A Temporal-Abstraction Rule Language for Medical Databases. Proceedings of the workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP 2003); 2003; Protaras, Cyprus; 2003.

67. Starren J, Xie G. Comparison of three Knowledge Representation formalisms for encoding the NCEP Cholesterol Guidelines. Proc Annu Symp Comput Appl Med Care. 1994:792-6.

68. Bellazzi R, Magni P, Larizza C, De Nicolao G, Riva A, Stefanelli M. Mining biomedical time series by combining structural analysis and temporal abstractions. Proc AMIA Symp. 1998:160-4.

69. Kahn MG. Modeling time in medical decision-support programs. Med Decis Making. 1991 Oct-Dec;11(4):249-64.

70. Haimowitz IJ. Knowledge-Based Trend Detection and Diagnosis [Doctoral]. Cambridge, MA: Massachusetts Institute of Technology; 1994.

71. Kohane I. Temporal Reasoning in Medical Expert Systems. In: Salaman R, Blum B, Jorgensen M, editors. MEDINFO 86; 1986; Washington, DC: North-Holland; 1986. p. 170-4.

72. Haimowitz IJ, Kohane IS. Automated trend detection with alternate temporal hypotheses. IJCAI 93. 1993:146-51.

73. Haimowitz IJ. Intelligent diagnostic monitoring using trend templates. Proc Annu Symp Comput Appl Med Care. 1994:702-8.

74. Jones RW, Lowe A, Harrison MJ. Trend recognition in clinical signals using template-based methods. Methods Inf Med. 2000 Jun;39(2):101-4.

75. Chatfield C. Analysis of Time Series. Third ed. New York: Chapman and Hall; 1984.

76. Gall W, Duftschmid G, Dorda W. Moving time window aggregates over patient histories. Int J Med Inform. 2001 Oct;63(3):133-45.

77. Perng C-S, Parker DS. Temporal Coupling Verification in Time Series Databases. J Intell Inf Syst. 2000;15:29-49.

78. Elmasri R, Navathe SB. Fundamentals of database systems. 3rd ed. New York: Addison-Wesley; 2000.

79. Hoppner F. Time series abstraction methods -- A Survey. Workshop on Knowledge Discovery in Databases; 2002; 2002. p. 777-86.

80. Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. In: Waibel A, Lee K-F, editors. Readings in Speech Recognition. San Mateo, CA: Morgan Kauffmann; 1990. p. 159-65.

81. Fritsche L, Schlaefer A, Budde K, Schroeter K, Neumayer HH. Recognition of critical situations from time series of laboratory results by case-based reasoning. J Am Med Inform Assoc. 2002 Sep-Oct;9(5):520-8.

82. Li J, Leong TY. Using linear regression functions to abstract high-frequency data in medicine. Proc AMIA Symp. 2000:492-6.

83. Heckerman D. Probabilistic Interpretations for Mycin's Certainty Factors. In: Kanal L, Lemmer J, editors. Uncertainty in Artificial Intelligence. Amsterdam: Elsevier; 1986. p. 167-96.

84. Charniak E. Bayesian networks without tears. AI Magazine. 1991;12(4):50-63.

85. Aliferis CF, Cooper GF. Temporal representation design principles: an assessment in the domain of liver transplantation. Proc AMIA Symp. 1998:170-4.

86. Aliferis CF, Cooper GF, Pollack ME, Buchanan BG, Wagner MM. Representing and developing temporally abstracted knowledge as a means towards facilitating time modeling in medical decision-support systems. Comput Biol Med. 1997 Sep;27(5):411-34.

87. Ho TB, Nguyen DT, Kawasaki S, Le SQ, Nguyen DD, Yokoi H, et al. Mining Hepatitis Data with Temporal Abstraction. In: Domingos P, Faloutsos C, Senator T, Kargupta H, Getoor L, editors. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2003; 2003; Washington, DC, USA: The Association for Computing Machinery; 2003. p. 369-77.

88. Barnett BJ, Wickens CD. Display proximity in multicue information integration: The benefits of boxes. Hum Factors. 1988;30(1):15-24.

89. Colman AM, Stirk JA. Singleton bias and lexicographic preferences among equally valued alternatives. Journal of Economic Behavior & Organization. 1999 1999/12;40(4):337-51.

90. Willard KE, Connelly DP, Johnson JR. Radical improvements in the display of clinical microbiology results: a Web-based clinical information system. Am J Med. 1996 Nov;101(5):541-9.

91. Wickens CD, Carswell CM. The proximity compatibility principle: Its psychological function and relevance to display design. Hum Factors. 1995;37:473-94.

92. Nygren E, Wyatt JC, Wright P. Helping clinicians to find data and avoid delays. Lancet. 1998 Oct 31;352(9138):1462-6.

93. Nygren E, Henriksson P. Reading the medical record. I. Analysis of physicians' ways of reading the medical record. Comput Methods Programs Biomed. 1992 Sep-Oct;39(1-2):1-12.

94. Benbasat I, Dexter A. An experimental evaluation of graphical and color-enhanced information systems under varying time constraints. Man Inf Sys Quart. 1986;10:59-81.

95. Lohse GL. The role of working memory on graphical information processing. Behav Inf Technol. 1997;16(6):297-308.

96. Cole WG, Stewart JG. Human performance evaluation of a metaphor graphic display for respiratory data. Methods Inf Med. 1994 Oct;33(4):390-6.

97. Cleveland WS, McGill R. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. J Am Stat Assoc. 1984 September;79(387):531-54.

98. Pachella RG, Somers P, Hardzinski M. A Psychophysical Approach to Dimensional Integrality. In: Getty DJ, Howard JH, Jr., editors. Auditory and Visual Pattern Recognition. Hillsdale, NJ: Lawrence Erlbaum Associates; 1981. p. 107-26.

99. Politser PE. How to make laboratory information more informative. Clin Chem. 1986 Aug;32(8):1510-6.

100. Shahar Y, Boaz D, Tahan G, Galperin M, Goren-Bar D, Kaizer H, et al. A Web-Based System for Interactive Visualization and Exploration of Time-oriented Clinical Data and Their Abstractions. Proc AMIA Annu Fall Symp. 2003:1073.

101. Cousins SB, Kahn MG. The visual display of temporal information. Artif Intell Med. 1991;3:341-57.

102. Plaisant C, Milash B, Rose A, Widoff S, Shneiderman B. LifeLines: visualizing personal histories. Proceedings of the SIGCHI conference on Human factors in computing systems: common ground. Vancouver, British Columbia, Canada: ACM Press; 1996. p. 221-7.

103. Bui AA, Taira RK, El-Saden S, Dordoni A, Aberle DR. Automated medical problem list generation: towards a patient timeline. Medinfo. 2004;11(Pt 1):587-91.

104. Patel VL, Arocha JF, Kaufman DR. A primer on aspects of cognition for medical informatics. J Am Med Inform Assoc. 2001 Jul-Aug;8(4):324-43.

105. Kushniruk AW, Patel VL. Cognitive evaluation of decision making processes and assessment of information technology in medicine. Int J Med Inf. 1998 Aug-Sep;51(2-3):83-90.

106. Cimino JJ, Patel VL, Kushniruk AW. Studying the human-computer-terminology interface. J Am Med Inform Assoc. 2001 Mar-Apr;8(2):163-73.

107. Patel VL, Kushniruk AW, Yang S, Yale JF. Impact of a computer-based patient record system on data collection, knowledge organization, and reasoning. J Am Med Inform Assoc. 2000 Nov-Dec;7(6):569-85.

108. Patel VL, Arocha JF, Kushniruk AW. Patients' and physicians' understanding of health and biomedical concepts: relationship to the design of EMR systems. J Biomed Inform. 2002 Feb;35(1):8-16.

109. Nielsen J. Usability Engineering. San Francisco: Morgan Kaufmann; 1993.

110. Kushniruk AW, Patel VL. Cognitive computer-based video analysis: its application in assessing the usability of medical systems. Medinfo. 1995;8 Pt 2:1566-9.

111. Regan M, Forsman R. The impact of the laboratory on disease management. Dis Manag. 2006 Apr;9(2):122-30.

112. Forsman R. The electronic medical record: implications for the laboratory. Clin Leadersh Manag Rev. 2000 Nov-Dec;14(6):292-5.

113. Becich MJ, Gilbertson JR, Gupta D, Patel A, Grzybicki DM, Raab SS. Pathology and patient safety: the critical role of pathology informatics in error reduction and quality initiatives. Clin Lab Med. 2004 Dec;24(4):913-43, vi.

114. Benge H, Bodor GS, Younger WA, Parl FF. Impact of managed care on the economics of laboratory operation in an academic medical center. Arch Pathol Lab Med. 1997 Jul;121(7):689-94.

115. van Walraven C, Goel V, Chan B. Effect of population-based interventions on laboratory utilization: a time-series analysis. Jama. 1998 Dec 16;280(23):2028-33.

116. Kisabeth RM. Laboratory adaptations--changing expectations. Clin Chem. 2001 Aug;47(8):1509-15.

117. Connelly DP, Lasky LC, Keller RM, Morrison DS. A system for graphical display of clinical laboratory data. Am J Clin Pathol. 1982 Nov;78(5):729-37.

118. Leape LL, Bates DW, Cullen DJ, Cooper J, Demonaco HJ, Gallivan T, et al. Systems analysis of adverse drug events. ADE Prevention Study Group. Jama. 1995 Jul 5;274(1):35-43.

119. Kraus DM, Calligaro IL, Hatoum HT. Multilevel model to assess appropriateness of pediatric serum drug concentrations. Am J Dis Child. 1991 Oct;145(10):1171-5.

120. Lesar TS, Briceland L, Stein DS. Factors related to errors in medication prescribing. Jama. 1997 Jan 22-29;277(4):312-7.

121. Politser PE. Intelligent display of laboratory data. SCAMC. 1984:402-5.

122. Hoeke JO, Gelsema ES, Wulkan RW, Leijnse B. Graphical non-linear representation of multi-dimensional laboratory measurements in their clinical context. Methods Inf Med. 1991 Apr;30(2):138-44.

123. Verheij R, Hoeke JO, Bonke B, van Strik R, Gelsema ES. Evaluation of techniques for the presentation of laboratory data. II: Accuracy of interpretation. Methods Inf Med. 1997 Jan;36(1):17-9.

124. Verheij R, Hoeke JO, Bonke B, van Strik R, Gelsema ES. Evaluation of techniques for the presentation of laboratory data. I: Time needed for interpretation. Methods Inf Med. 1997 Jan;36(1):11-6.

125. Stanton WM, Tang PC. Knowledge-based support for a physician's workstation. Proc Annu Symp Comput Appl Med Care. 1991:649-53.

126. Tang PC, Annevelink J, Suermondt HJ, Young CY. Semantic integration of information in a physician's workstation. Int J Biomed Comput. 1994 Feb;35(1):47-60.

127. Tang PC, Annevelink J, Fafchamps D, Stanton WM, Young CY. Physicians' workstations: integrated information management for clinicians. Proc Annu Symp Comput Appl Med Care. 1991:569-73.

128. Pauker SG. Clinical Decision Making: Handling and Analyzing Clinical Data. In: Bennett JC, Plum F, editors. Cecil Textbook of Medicine. 20th ed. Philadelphia: W. B. Saunders Company; 1996. p. 78-83.

129. Famili A, Shen W-M, Weber R, Simoudis E. Data Preprocessing and Intelligent Data Analysis. Intelligent Data Analysis. 1997;1(1):3-23.

130. Boaz D, Shahar Y. A framework for distributed mediation of temporal-abstraction queries to clinical databases. Artif Intell Med. 2005 May;34(1):3-24.

131. Zhang J, Tsui FC, Wagner MM, Hogan WR. Detection of outbreaks from time series data using wavelet transform. Proc AMIA Symp. 2003:748-52.

132. David J-M, Krivine J-P, Simmons R. Second Generation Expert Systems: A Step Forward in Knowledge Engineering. In: David J-M, Krivine J-P, Simmons R, editors. Second generation expert systems. Berlin ; New York: Springer-Verlag; 1993. p. 3-23.

133. Shahar Y, Chen H, Stites DP, Basso LV, Kaizer H, Wilson DM, et al. Semi-automated entry of clinical temporal-abstraction knowledge. J Am Med Inform Assoc. 1999 Nov-Dec;6(6):494-511.

134. Musen MA. Dimensions of knowledge sharing and reuse. Comput Biomed Res. 1992 Oct;25(5):435-67.

135. Jha AK, Kuperman GJ, Teich JM, Leape L, Shea B, Rittenberg E, et al. Identifying adverse drug events: development of a computer-based monitor and comparison with chart review and stimulated voluntary report. J Am Med Inform Assoc. 1998 May-Jun;5(3):305-14.

136. Chaudhry SI, Olofinboba KA, Krumholz HM. Detection of errors by attending physicians on a general medicine service. J Gen Intern Med. 2003 Aug;18(8):595-600.

137. Haimowitz IJ, Kohane IS. An epistemology for clinically significant trends. AAAI. 1993:176-81.

138. Haimowitz IJ, Le PP, Kohane IS. Clinical monitoring using regression-based trend templates. Artif Intell Med. 1995 Dec;7(6):473-96.

139. Shahar Y, Musen MA. RESUME: a temporal-abstraction system for patient monitoring. Comput Biomed Res. 1993 Jun;26(3):255-73.

140. Weydert JA, Nobbs ND, Feld R, Kemp JD. A simple, focused, computerized query to detect overutilization of laboratory tests. Arch Pathol Lab Med. 2005 Sep;129(9):1141-3.

141. Ridgeway JJ, Weyrich DL, Benedetti TJ. Fetal heart rate changes associated with uterine rupture. Obstet Gynecol. 2004 Mar;103(3):506-12.

142. Snodgrass R, Bohlen MH, Jensen CS, Steiner A. Transitioning Temporal Support in TSQL2 to SQL3. In: Etzion O, Jajodia S, Sripada S, editors. Temporal Databases: Research and Practice. Berlin, NY: Springer; 1998. p. 150-94.

143. Karadimas HC, Chailloleau C, Hemery F, Simonnet J, Lepage E. Arden/J: an architecture for MLM execution on the Java platform. J Am Med Inform Assoc. 2002 Jul-Aug;9(4):359-68.

144. Sherman EH, Hripcsak G, Starren J, Jenders RA, Clayton P. Using intermediate states to improve the ability of the Arden Syntax to implement care plans and reuse knowledge. Proc Annu Symp Comput Appl Med Care. 1995:238-42.

145. O'Connor MJ, Tu SW, Musen MA. The Chronus II Temporal Database Mediator. Proc AMIA Symp. 2002:657-571.

146. Sibai BM, Barton JR. Dexamethasone to improve maternal outcome in women with hemolysis, elevated liver enzymes, and low platelets syndrome. Am J Obstet Gynecol. 2005 Nov;193(5):1587-90.

147. Fonseca JE, Mendez F, Catano C, Arias F. Dexamethasone treatment does not improve the outcome of women with HELLP syndrome: a double-blind, placebo-controlled, randomized clinical trial. Am J Obstet Gynecol. 2005 Nov;193(5):1591-8.

148. Martin JN, Jr., Rinehart BK, May WL, Magann EF, Terrone DA, Blake PG. The spectrum of severe preeclampsia: comparative analysis by HELLP (hemolysis, elevated liver enzyme levels, and low platelet count) syndrome classification. Am J Obstet Gynecol. 1999;180(6):1373-84.

149. Yan J, Fine J. Estimating equations for association structures. Stat Med. 2004 Mar 30;23(6):859-74.

150. Winer BJ. Statistical principles in experimental design. New York: McGraw-Hill; 1971.

151. Grudin J. The case against user interface consistency. Commun ACM. 1989 October;32(10):1164-73.

152. Lankford C. Gazetracker: software designed to facilitate eye movement analysis. Proceedings of the 2000 symposium on Eye tracking research & applications. Palm Beach Gardens, Florida, United States: ACM Press; 2000.

153. Friedman CP, Wyatt J, Shortliffe EH. Evaluation methods in medical informatics. New York: Springer; 1997.

154. Mitchell TM. Machine learning and data mining. Commun ACM. 1999 Nov 1999;42(11):30-6.

155. Nigrin DJ, Kohane IS. Data mining by clinicians. Proc AMIA Symp. 1998:957-61.

156. Kohn LT, Corrigan JM, Donaldson MS, editors. To Err is Human: Building a Safer Health System. Washington, D. C.: National Academy Press; 1999.

157. Reason J. Human error: models and management. Bmj. 2000 Mar 18;320(7237):768-70.

158. Price CP. Point of care testing. Bmj. 2001 May 26;322(7297):1285-8.

159. Kost GJ. Preventing medical errors in point-of-care testing: security, validation, safeguards, and connectivity. Arch Pathol Lab Med. 2001 Oct;125(10):1307-15.

160. Kilgore ML, Steindel SJ, Smith JA. Continuous quality improvement for point-of-care testing using background monitoring of duplicate specimens. Arch Pathol Lab Med. 1999 Sep;123(9):824-8.

161. Bedside blood glucose monitoring in hospitals. Diabetes Care. 2004 Jan;27 Suppl 1:S104.