Model-Based Visualization of Temporal Abstractions

Yuval Shahar and Cleve Cheng

Stanford Medical Informatics
251 Campus Drive, Medical School Office Building (MSOB) x215
Stanford University, Stanford, CA 94305-5479, USA
email: shahar,Ccheng@smi.stanford.edu

Address and details of Corresponding Author:

Yuval Shahar, M.D., Ph.D.
Assistant Professor of Medicine and Computer Science
Stanford Medical Informatics
Medical School Office Building x215
251 Campus Drive
Stanford University
Stanford, CA 94305-5479
fax: +1-650-725-7944
tel.: +1-650-725-3393
email: shahar@smi.stanford.edu
WWW: http://smi.stanford.edu/people/shahar/index.html

Suggested Running Title: Model-Based Visualization of Temporal Abstractions

Abstract

We describe a new conceptual methodology and related computational architecture called Knowledge-based Navigation of Abstractions for Visualization and Explanation (KNAVE). KNAVE is a domain-independent framework specific to the task of interpretation, summarization, visualization, explanation, and interactive exploration, in a context-sensitive manner, of time-oriented raw data and the multiple levels of higher-level, interval-based concepts that can be abstracted from these data. The KNAVE domain-independent exploration operators are based on the relations defined in the knowledge-based temporal-abstraction problem-solving method, which is used to abstract the data, and thus can directly use the domain-specific knowledge base on which that method relies. Thus, the domain-specific semantics are driving the domain-independent visualization and exploration processes, and the data is viewed through a filter of domain-specific knowledge. By accessing the domain-specific temporal-abstraction knowledge base and the domain-specific time-oriented database, the KNAVE modules enable users to query for domain-specific temporal abstractions and to change the focus of the visualization, thus reusing for a different task (visualization and exploration) the same domain model acquired for abstraction purposes. We focus here on the methodology, but also describe a preliminary evaluation of the KNAVE prototype in a medical domain. Our experiment incorporated seven users, a large medical patient record, and three complex temporal queries, typical of guideline-based care, that the users were required to answer and/or explore. The results of the preliminary experiment have been encouraging. The new methodology has potentially broad implications for planning, monitoring, explanation, and interactive data mining of time-oriented data.

Keywords: temporal reasoning; temporal abstraction; information visualization; exploration; data mining

1 TEMPORAL ABSTRACTION AND INFORMATION VISUALIZATION

Many domains of human endeavor and scientific research require the collection of substantial numbers of data over time, and the abstraction of those data into higher-level concepts and patterns meaningful for that domain, a task we call **temporal abstraction**. It is impossible to consider data trends, evolution of processes, or causality without an explicit representation of time. This is especially true in medical domains, such as when monitoring and planning therapy for patients who have chronic diseases (e.g., cancer, AIDS, or diabetes) (Figure 1).

Visualization and exploration of information in general, and of large amounts of time-oriented data in particular, is essential for effective decision making. Examples in medical domains include deciding, during therapy by a particular clinical guideline, whether a patient, who had previous bone-marrow transplantation, had more than two episodes of bone-marrow toxicity of severity grade 1 or higher, each lasting at least a week. The severity classification is often specific to the context of therapy by a particular clinical guideline, which includes a drug that might be toxic to the bone marrow (see Figure 1). Examples in engineering domains include deciding whether a particular traffic control action, such as diverting traffic in a certain congested highway zone, has been effective, as judged by the resulting observed pattern of traffic over time, compared to the pattern expected in that context. Thus, effective visualization often requires preprocessing to abstract the data, using domain-specific knowledge, into meaningful interpretations.

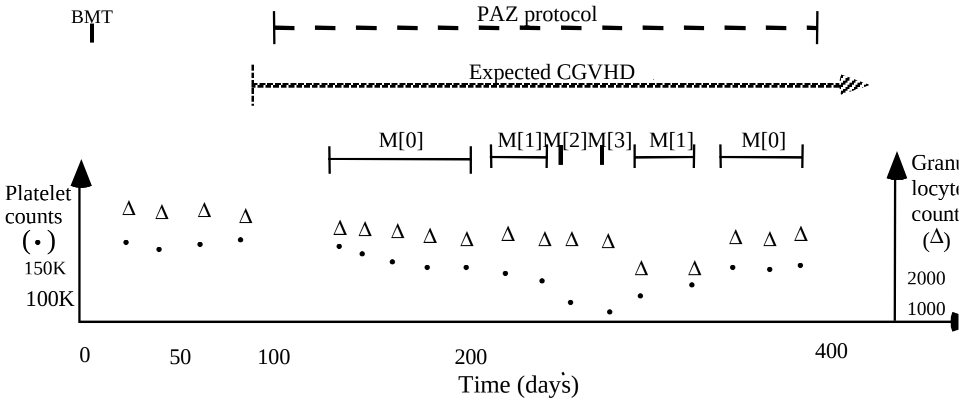


Figure 1: Temporal abstraction in a medical domain. Raw data are plotted over time at the bottom. External events and the abstractions computed from the data are plotted as intervals above the data. BMT = a bone-marrow transplantation event; PAZ = a therapy protocol for treating chronic graft-versus-host disease (CGVHD), a complication of BMT; ⊢ ⊣ = event; • = platelet counts; Δ = granulocyte counts; ⊢→ = context interval; ⊢⊣ = abstraction interval; M[*n*] = bone-marrow–toxicity grade *n*.

Larkin and Simon (1987) have demonstrated that the usefulness of visual representations is mainly due to (1) reduction of logical computation through the use of direct perceptual inference, and (2) reduction of necessary search for information through the use of efficient graphical representations. Indeed, we have noted in the past (Shahar and Musen, 1996) that experts in therapy of diabetes preferred graphical charts over data tables for the purpose of noticing meaningful diurnal trends in blood-glucose values and insulin administrations.

The ability to automatically create, visualize, and explore interval-based abstractions of time-stamped data in context-sensitive manner in multiple domains, such as in various medical domains, has several useful implications:

1. *Data summaries* of time-oriented electronic data have an immediate value to a human user, such as to a physician scanning a long patient record for meaningful trends.
2. Machine-readable temporal abstractions support *recommendations* by intelligent decision-support systems.

3. Abstractions support *monitoring* of plans, such as therapy plans, during execution.
4. Meaningful time-oriented contexts that are induced from the data enable generation of *context-specific abstractions* and maintenance of *several interpretations* of the same data within different contexts.
5. Temporal abstractions are helpful for *explanation* of recommended actions by an intelligent system.
6. Temporal abstractions are a useful *representation of intentions* of plans, such as goals of clinical guidelines.
7. Visualization and exploration of time-oriented data and its multiple-level abstractions supports a form of manual *interactive data mining* of time-oriented databases.

Thus, it is highly desirable to assist users who need to understand large amounts of time-oriented data, such as physicians who monitor patients over significant periods, by providing a useful visualization of these data and of their abstractions. The abstractions being visualized must be specific to the context in which the data were acquired, and should include significant amounts of domain-specific interpretation, thus reducing considerably the information overload on the user. To further reduce the information overload, succinct summaries must include not only abstractions of the data that hold over time *points,* such as "moderate anemia at 9:00 a.m. on January 5 1997," but also conclusions that hold over time *intervals*, such as "5 months of decreasing liver-enzyme levels in the context of recovering from hepatitis." Finally, domain-specific abstraction, visualization and exploration of time-oriented data and meaningful concepts implied by these data requires acquisition and disciplined maintenance of knowledge of domain-specific time-oriented properties of the particular data. Examples of such domain-specific knowledge include meaningful classifications into a more abstract temporal patterns, knowledge of whether similar patterns can be joined or should be considered as separate episodes, and an indication of how data should be best displayed over time.

One way of offering a data abstraction, visualization, and exploration service to users, which we propose in this paper, is to generate automatically short, informative, context-sensitive summaries of the data, at various levels of abstraction (see Figure 1), and to enable users to query, visualize, and explore these abstractions interactively.

In this paper, we discuss in depth a specific domain-independent conceptual methodology and a related software architecture that we have designed, which embody this philosophy. Our approach is based on our previous theoretical and practical development of a domain-independent method that performs automated abstraction of raw time-oriented data into meaningful concepts, using access to a declarative representation of domain-specific knowledge. We are also exploiting insights that we have gained from that research and from other studies we performed, regarding the appropriate organization and acquisition of the domain-specific knowledge that is needed for performance of the temporal-abstraction task by our method. Finally, we are capitalizing on several additional computational modules that we have developed previously, such as for mediation of temporal queries to a time-oriented database.

This paper focuses on (1) our abstraction and visualization methodology, and in particular, on the semantics of its interactive exploration operators, and (2) the computational architecture that is implied by our methodology. We do *not* aim in this paper to either justify or evaluate a particular graphical display. However, we will demonstrate and clarify our ideas regarding knowledge-based interactive exploration by using a preliminary software prototype that we have designed and that implements the core functionality of our architecture. We will also present a developmental evaluation of that prototype in a medical domain, to demonstrate the feasibility of our approach.

In the following sections, we start by presenting the knowledge-based visualization and exploration architecture. We then explain briefly the temporal-abstraction methodology and computational infrastructure underlying that architecture; we have discussed these components at length in previous publications. We then describe in depth the new query, visualization, and exploration methodology, using our prototype for demonstration purposes. We briefly describe a preliminary evaluation of that prototype that we have performed within a small group of users. We then discuss our work in the context of previous research pertinent to the several subtasks performed by our architecture. We conclude with a note on our current and future work.

2 THE KNAVE ARCHITECTURE

Our preliminary discussion of the task of knowledge–based abstraction, visualization and exploration of time-oriented abstractions highlights that fact that performance of this task requires an integrated solution to three subtasks: *temporal abstraction*, *knowledge acquisition*, and *information visualization*. In particular, we would claim that an architecture offering a knowledge-based solution to the temporal visualization and exploration task must include three components: (1) The time-oriented database (and a method to access it effectively); (2) a knowledge-base of domain-specific properties; and (3) a module for display and exploration of the data. Furthermore, an appropriate solution that addresses the requirements mentioned in Section 1 must also consider the distributed nature of time-oriented databases (e.g., patient databases), knowledge bases, and problem-solving processes. Bearing all these logical considerations in mind, we have designed an architecture, specific to discussion of the task of abstraction, visualization and exploration of time-oriented data and its abstractions, that capitalizes on recent advances in the understanding of all three tasks. We refer to our architecture as **Knowledge-based Exploration of Abstractions for Visualization and Explanation** (**KNAVE**). The KNAVE architecture is presented in Figure 2.

The KNAVE architecture comprises three types of modules, including several components (see Figure 2). Several of these components (in particular, the temporal-abstraction module and knowledge-acquisition tool) have been topics of our previous research.

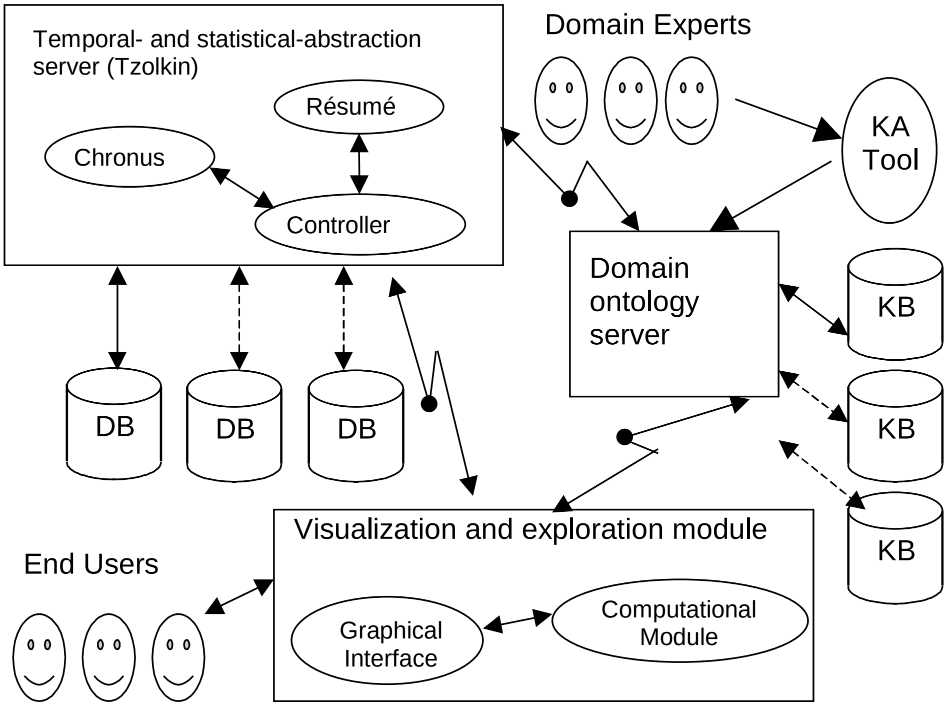


Figure 2: The KNAVE architecture. End users interact with the visualization and exploration module to ask temporal queries. These queries are answered by the temporal- and statistical-abstraction service, using domain-specific data from the appropriate database, and temporal-abstraction knowledge from the appropriate knowledge base (ontology). The visualization and exploration module allows users to explore the resultant abstractions, using temporal-abstraction and visualization knowledge from the domain's ontology. Arrow direction indicates data flow. DB = database; KB = knowledge base; KA = knowledge-acquisition.

In the KNAVE architecture, users (e.g., physicians) send complex temporal and statistical queries at varying levels of abstraction from local workstations (through the visualization module) to the **temporal- and statistical-abstraction** (**TSA**) **server**. The TSA server is a *database mediator* (Wiederhold, 1992) that accesses multiple domain-specific time-oriented databases to provide answers to the queries, using the **RÉSUMÉ** knowledge-based temporal-abstraction system, which we have developed in previous research. Typically, time-oriented databases include only raw time-oriented data, and no abstractions of those data; examples include databases of electronic medical records. To interpret the data correctly, the TSA server uses knowledge about temporal properties of the data by accessing the appropriate domain-specific knowledge base (through the ontology server). Thus, the TSA server is a *temporal mediator* to the time-oriented databases. To build the TSA server, we are using the **Tzolkin** temporal mediator, also developed in our previous research (Nguyen et al., in press). Domain experts maintain the domain-specific temporal-abstraction knowledge bases using graphical knowledge-acquisition tools that we have developed. Domain-specific knowledge includes visualization knowledge and user preferences as well as temporal-abstraction knowledge. The domain experts, the TSA server, and the local visualization and exploration processes access the domain-specific knowledge bases through the **domain ontology server**. The visualization and exploration client is an innovative computational module that we have developed recently. The visualization and exploration module is intended for use by all target users (e.g., physicians who need to summarize and explore patient data); it coordinates all of the distributed services. The visualization and exploration module includes two components: (1) a computational module, which processes information obtained from the TSA and domain-ontology servers to support tasks such as interactive semantic exploration (e.g., “From what data is this bone-marrow-toxicity pattern abstracted?”), dynamic explanation (e.g., “What classification knowledge was used to create this abstraction?”) and dynamic hypothetical exploration (e.g., “What would the abstract pattern look like if the patient’s platelet count were, in fact, lower on 1/6/98?”), and (2) a user interface module, which displays the various windows and widgets that make up the KNAVE display interface and which drive the computational component. The computational component filters the information coming from the TSA and ontology servers to construct an internal representation which is efficient for the task of interactive exploration. Thus, the representation of temporal abstractions and relations between them is separate from the process of displaying the information.

2.1 The Knowledge-Based Visualization and Exploration Method

The methodology underlying KNAVE is based on our previous work on the **knowledge-based temporal-abstraction method** (Shahar, 1997). This method performs the temporal-abstraction task (see Section 1). The knowledge-based temporal-abstraction method depends on four well-defined domain-specific **knowledge types**: structural, classification (functional), temporal-semantic (logical), and temporal-dynamic (probabilistic) knowledge. Values for the four knowledge types are specified within the domain's **temporal-abstraction ontology** (which includes properties of measurable parameters, external actions, and contexts induced by parameters or actions) when developing a temporal-abstraction system, and are acquired from domain experts through a specialized graphical knowledge-acquisition tool. The knowledge-based temporal-abstraction method has been implemented in the **RÉSUMÉ** system (Shahar and Musen, 1993), and has been evaluated in multiple medical (Shahar and Musen, 1996) and engineering (Shahar and Molina, 1998) domains with encouraging results.

Although all of the computational visualization and exploration operators used in the KNAVE system are domain independent, these operators are mapped to the domain-specific knowledge maintained by the ontology server. That knowledge, in turn, has been acquired from domain experts for the purpose of abstraction of the domain-dependent data over time. Thus, the domain-specific semantics are driving the visualization and exploration processes. These semantics are defined by the ontology of the knowledge-based temporal-abstraction method.

The KNAVE framework enables users to query for domain-specific abstractions and to change the focus of the visualization in a natural manner, by reusing the same domain model that has been acquired from domain experts to perform the temporal-abstraction task. Typically, that task is a part of planning, monitoring, data interpretation, or other applications. The KNAVE framework's interactive-exploration operators reuse the well-defined, domain-independent semantic links in the domain's temporal-abstraction ontology, thus defining a conceptual **knowledge-based visualization and exploration method**. In the knowledge-based visualization and exploration method, exploration of predefined relations in the *domain's (temporal-abstraction) ontology* causes certain computational transformations in the *displayed abstractions*. For example, exploration of a functional-dependency link related to a visualized abstraction leads to the display of the data and/or abstractions from which the visualized abstraction is derived. Different combinations of parallel motion along several semantic links in the domain's temporal-abstraction ontology define precisely several **semantic-zoom operators**. These operators are augmented by purely syntactic visualization operators, such as overlay, magnification, and temporal scrolling.

A rather unique feature of exploring time-oriented data and temporal abstractions is that each type of data or abstraction has certain *ranges of temporal granularit*y within which it is meaningful to visualize it (e.g., hours to days) and beyond which it should be visualized differently by using some domain-specific *aggregation operator* (e.g., mean, standard deviation, distribution, etc). The KNAVE framework enables users to change on the fly the temporal granularity level of the interface (i.e., the relevant resolution of the presentation and interpretation process, as opposed to a syntactic zoom, which involves only graphical magnification of the data).

As we show when discussing the explanation operators, the KNAVE framework provides several mechanisms for explanations of data interpretations in a natural fashion by capitalizing on the direct access to the domain's temporal-abstraction ontology (e.g., zooming on the data from which a pattern is derived, or displaying dynamically the classification knowledge relevant to the creation of an abstraction, by retrieving it from the domain's ontology).

Finally, the KNAVE architecture lends itself well to explicit representation of domain-specific and individual-user preference models. These models can be customized at the level of either the graphical display or the semantics of the knowledge-based exploration model.

Thus, the data- and knowledge-flow process inherent in the KNAVE architecture can be summarized as follows:

1. The knowledge-based temporal-abstraction methodology is used to acquire, represent, and maintain an explicit representation of domain-specific temporal-abstraction knowledge as part of the domain's ontology, using semi-automated graphical knowledge-acquisition tools.
2. The computational temporal-abstraction mechanisms in the TSA server create domain-specific, context-sensitive temporal abstractions of the data.
3. At runtime, direct access is provided to the domain's temporal-abstraction ontology. Direct access supports the formulation of the user's initial queries, the creation of the resulting static temporal abstractions, and the dynamic exploration of these abstractions by moving along predefined ontological temporal-abstraction semantic links, such as ABSTRACTED-FROM, IS-A, or SUBCONTEXT. Thus, the user explores multiple levels of domain-specific abstractions in a domain-independent, but task-specific, manner, and the domain-specific ontology is driving the exploration process.
4. Additional temporal-browsing operators are provided to the user; several are purely syntactic and do not depend on domain knowledge, and several use the direct access to the domain's ontology, such as determination of which data or abstractions are relevant for visualization at certain temporal-granularity ranges, and what alternative representations are useful at other temporal resolution levels.

5. During the exploration process, the user can obtain context-sensitive explanations to questions such as “what classification function defines the creation of this abstraction from those data?” by exploiting the direct access to the domain’s temporal-abstraction ontology.

3 KNOWLEDGE-BASED TEMPORAL-ABSTRACTION

In this section, we briefly describe our previous work on the knowledge-based temporal-abstraction method, several of our theoretical and practical results, and the computational modules relevant to the KNAVE architecture that we have developed as part of that work. These theoretical results and computational components support directly several of the knowledge-representation, knowledge-acquisition, and computational requirements of the KNAVE methodology. Furthermore, the semantics of the knowledge-based temporal-abstraction method define, in turn, the semantics of the knowledge-based query, visualization, and, in particular, exploration operators. Thus, the KNAVE framework can be viewed as an innovative reuse of the temporal-abstraction ontology and of the computational mechanisms used by the knowledge-based temporal-abstraction method within a new conceptual and computational framework that supports a different task. That task consists of providing interactive visualization and exploration of time-stamped data and their multiple-level time-oriented abstractions.

The **temporal-abstraction task** is an *interpretation task*: given time-stamped parameters (raw and abstracted data), external events, and the user’s abstraction goals, produce time-interval–based abstractions of the data that interpret past and present states and trends and that are relevant for a given set of goals (Shahar, 1997) (see Figure 1). The objective of our previous research was to develop a theoretical framework and an accompanying technology that can solve the temporal-abstraction task and thus provide concise, informative, context-sensitive summaries of time-oriented data stored on electronic media, such as medical records. Predefined or ad hoc queries would be answered at various levels of abstraction. The output of such a tool, as explained in Section 1, is useful for multiple tasks, both to human users and to intelligent decision-support systems.

3.1 The Knowledge-Based Temporal-Abstraction Problem-Solving Method

The framework that we have developed for solving the temporal-abstraction task is an extension of our previous work on temporal-abstraction mechanisms (Shahar et al., 1992; Shahar and Musen, 1993). We have defined a general *problem-solving method* (Eriksson et al., 1995) for interpreting data in time-oriented domains, with clear semantics for both the problem-solving method and its domain-specific *knowledge* requirements: the **knowledge-based temporal-abstraction method** (Shahar, 1997). This method comprises a knowledge-level (Newell, 1982) representation of the temporal-abstraction task and the knowledge required to solve that task. The knowledge-based temporal-abstraction method has a formal model of input and output entities, of their relations, and of properties associated with these entities (Shahar, 1997).

The knowledge-based temporal-abstraction method decomposes the temporal-abstraction task into five parallel *subtasks*: temporal context restriction, vertical temporal inference, horizontal temporal inference, temporal interpolation, and temporal pattern matching. The five subtasks of the knowledge-based temporal-abstraction *method* are solved respectively by five temporal-abstraction *mechanisms* (nondecomposable computational modules).

The temporal-abstraction mechanisms produce output abstractions of several abstraction types: **state** (e.g., LOW), **gradient** (e.g., INCREASING), **rate** (e.g., FAST), and **pattern** (e.g., QUIESCENT-ONSET).

The **context-forming mechanism** creates **context intervals** over which hold **interpretation contexts** (Shahar, 1998). Context intervals create a relevant *frame of reference* for interpretation and enable the temporal-abstraction mechanisms to focus only on abstractions relevant for particular contexts, thus creating interpretations that are context-specific and avoiding unnecessary computations. Interpretation contexts are *induced* dynamically at runtime by the presence of a *context-forming proposition*, not necessarily with the same temporal scope (e.g., the event of administration of AZT might create a delayed, future interpretation context of potential toxicity due to AZT).

The **contemporaneous-abstraction mechanism** abstracts one or more parameters and their values, attached to contemporaneous time points or time intervals, into a value of a new, abstract parameter. Thus, it performs a classification of a given parameter's value or a computational transformation of the values of several parameters, using a classification function. An ABSTRACTED-INTO relation exists between input and output parameters.

The **temporal-inference mechanism** performs two subtasks: (1) inference of specific types of interval-based logical conclusions, given interval-based propositions, using a deductive extension of Shoham's temporal-semantic properties (Shoham, 1987) (e.g., unlike two consecutive periods of anemia, two episodes of 9-month pregnancies cannot be summarized as an episode of an 18-month pregnancy, since they are not *concatenable*, a temporal-semantic property (Shoham, 1987)), and (2) determination of the domain value of an abstraction created from two meeting abstractions (e.g., for a gradient abstraction, DECREASING ⊕ SAME = NONINCREASING (Shahar et al., 1992)).

The **temporal-interpolation mechanism** bridges gaps between *temporally-disjoint* propositions of similar types, using domain-specific temporal-dynamic knowledge about the dynamic behavior of the parameters involved (Shahar, 1997). The temporal-interpolation mechanism uses *local and global truth-persistence functions* to join temporally disjoint abstractions when values for direct determination of the abstractions are missing (Shahar, 1999).

The **temporal-pattern–matching mechanism** matches predefined linear (sequential) and periodic (cyclical or episodic) temporal patterns, as well as runtime temporal queries, with data, concluded abstractions or patterns, and external events. The output is a *pattern abstraction*, which holds over an interval. Complex combinations of temporal and value constraints can be used to define linear and periodic patterns (Chakravarty and Shahar, 1999).

The temporal-abstraction mechanisms require four domain-specific knowledge types for any particular domain: (1) **structural knowledge** (e.g., ABSTRACTED-INTO relations); (2) **classification knowledge** (e.g., definition of a parameter range as LOW); (3) **temporal-semantic knowledge** (e.g., the *concatenable* property); and (4) **temporal-dynamic knowledge** (e.g., persistence of a proposition over time when data is unavailable).

The *input* to the temporal-abstraction task is a set of measured time-stamped **parameters** (e.g., temperature), external **events** (e.g., insulin injections), **abstraction goals** (e.g., diabetes therapy), and domain-specific temporal-abstraction knowledge. The *output* of the temporal-abstraction task is a set of interval-based, context-specific parameters or patterns at the same or at a higher level of abstraction and their respective values (e.g., "a period of 5 weeks of severe anemia in the context of therapy with AZT"). An abstraction of a parameter (e.g., the state of the hemoglobin-level) is also a parameter. *Time intervals* are constructed from pairs of *time stamps*; time points are zero-length intervals. We call the structure {<*parameter*, *value*, *context*>, *interval*} a **parameter interval**; it denotes that the **parameter proposition** "the *parameter* has a particular *value* given a specific *context* of interpretation" holds during a specific time interval. Propositions hold only over time intervals. If the parameter is an abstract (computed) parameter, such a structure is called an **abstraction**. A **pattern interval** is defined in similar fashion. Output abstractions or patterns hold within the *interpretation contexts* created by the context-forming mechanism.

A **temporal-abstraction ontology** includes (1) a **parameter ontology**—a theory of the relevant parameters and their temporal properties in the domain and the relations among these parameters (e.g., IS-A, ABSTRACTED-INTO), (2) a **pattern ontology**, which defines all patterns and their relations to other patterns and parameters (e.g., a linear pattern can have a COMPOSED-FROM relation to an abstraction, which for practical purposes can be viewed as an ABSTRACTED-INTO relation), (3) an **event ontology**, which includes external events (e.g. medications), their interrelations (e.g., PART-OF relations) and properties, (4) a **context ontology**, which includes interpretation contexts (e.g., the temporal context defined by the effect of a drug) and relations (e.g., SUBCONTEXT) among interpretation contexts, (5) an **abstraction-goal ontology**, which includes all abstraction goals (which can induce contexts; e.g., monitoring of diabetes therapy) and their IS-A relations; and (6) all relations between inducing propositions and induced contexts (e.g., INDUCED-CONTEXTS).

3.2 The RÉSUMÉ Knowledge-Based Temporal-Abstraction System

In our previous research, we have implemented the knowledge-based temporal-abstraction method as the **RÉSUMÉ** system (Shahar and Musen, 1993; Shahar and Musen, 1996) in the CLIPS shell (Giarratano and Riley, 1994). RÉSUMÉ generates temporal abstractions, given time-stamped data and events, and the domain's temporal-abstraction ontology. RÉSUMÉ is composed of a *temporal-reasoning module* (the five temporal-abstraction mechanisms), a static *temporal-abstraction domain knowledge base* (the temporal-abstraction ontology), and a *dynamic temporal fact base* that stores input intervals representing external events, abstractions, and raw data and output interval-based interpretation contexts and abstractions. Updates that cause retraction of previously concluded abstractions are propagated by a *truth-maintenance system* (Shahar and Musen, 1996). Apart from *conceptual* advantages, such as facilitation of knowledge-acquisition and maintenance, the RÉSUMÉ architecture has several *computational* advantages (Shahar and Musen, 1996; Shahar, 1997).

3.2.1 Application of The RÉSUMÉ System

As part of our previous research, we tested the RÉSUMÉ system in several different clinical and engineering domains: protocol-based care (experimental therapy of AIDS patients, therapy of chronic graft-versus-host disease, and prevention of AIDS-related complications) (Shahar and Musen, 1993; Shahar and Musen, 1993); monitoring of children’s growth (Kuilboer et al., 1993); therapy of patients who have insulin-dependent diabetes (Shahar and Musen, 1996); assessment of the quality of guideline-based care, when the intentions of the guideline-designers are expressed as temporal abstractions to achieve or avoid (Shahar et al., 1998); and even monitoring of traffic and evaluation of traffic-control actions (Shahar and Molina, 1998). We evaluated the feasibility of knowledge acquisition, representation, and maintenance, and applied the knowledge-based temporal-abstraction method to various test cases. We found that both general temporal-abstraction computational knowledge and domain-specific temporal-abstraction knowledge were reusable, and that the results correlated with abstractions created manually by domain experts. Furthermore, the experiments in the traffic-control domain emphasized the generality of our methodology and its potential applicability not only to *time*-oriented data, but also to abstraction of data measured over any *linear distance measure*, and in particular, over linear *space* (e.g., spatial abstraction of data from traffic sensors along a highway).

3.3 The Tzolkin Temporal Mediator

We have incorporated the RÉSUMÉ system within a domain-independent *temporal mediator* (Wiederhold 1992; Das et al., 1994), the **Tzolkin** system (Nguyen et al., in press). The Tzolkin system combines the RÉSUMÉ temporal-reasoning system with the **Chronus** temporal-maintenance system (Das and Musen, 1994), which can access and manage a time-oriented relational database. Tzolkin answers all temporal-abstraction queries referred to a time-oriented database by analyzing the query, retrieving the relevant data from the database and knowledge from the domain's temporal-abstraction ontology, and returning the appropriate abstractions. The Tzolkin system is now an integrated key component in the **EON** architecture for guideline-based medical care (Musen et al., 1996), which is being evaluated in domains such as guideline-based breast-cancer therapy, and is part of an industrial collaboration.

3.4 The Temporal-Abstraction Knowledge-Acquisition Tool

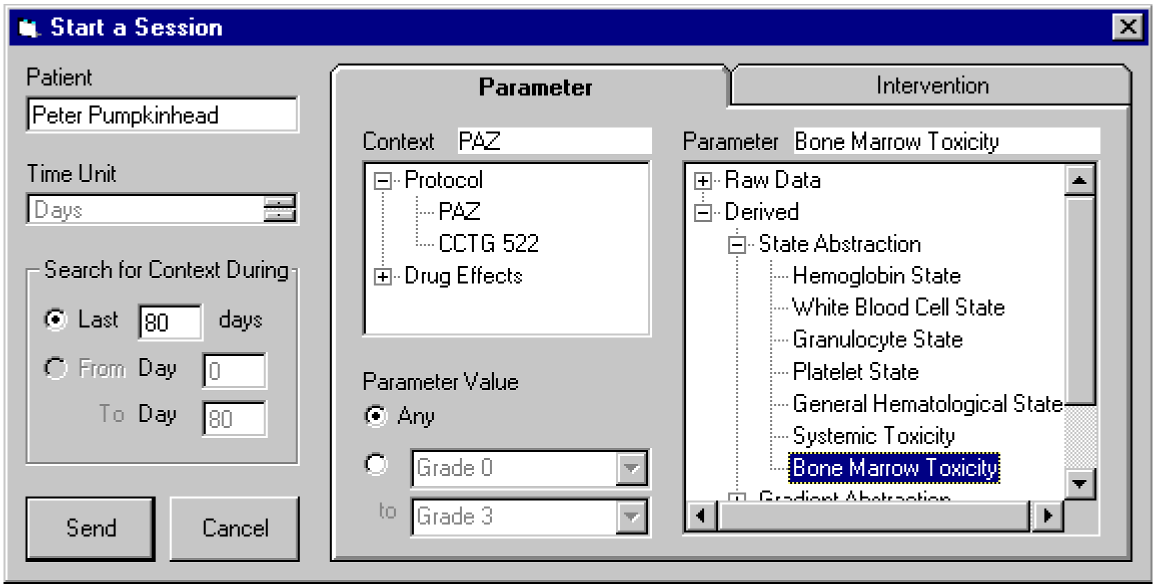
As part of our previous research, we have constructed a graphical knowledge-acquisition tool for automated acquisition of temporal-abstraction knowledge from domain experts (Stein et al., 1996), using tools from the Protégé project. The **Protégé** framework (Musen, 1992; Puerta et al., 1992; Musen et al., 1995; Tu et al., 1995; Musen, 1998) is a workbench for construction and use of ontologies, for creation of domain-specific knowledge-acquisition tools derived from ontologies, and for entering of domain knowledge into these knowledge-acquisition tools. One advantage of the Protégé approach is the production, given the relevant problem-solving-method and domain ontologies, of *automated knowledge-acquisition tools,* tailored for the selected problem-solving method and domain. Recent evaluation of the usability of the temporal-abstraction knowledge-acquisition tool, using experts from several different clinical domains, has been quite encouraging (Shahar et al., in press).

4 IMPLEMENTATION OF THE KNOWLEDGE-BASED VISUALIZATION AND EXPLORATION METHOD WITHIN THE KNAVE ARCHITECTURE

In this section, we explain the semantics of the knowledge-based visualization and exploration method by walking through an example in the domain of guideline-based care. The example is annotated by images from our prototype of the visualization and exploration module (originally developed in Visual Basic, and currently re-implemented in Java on a Windows/NT system). Our aim is not to claim any particular advantages to our display interface, but rather (1) to clarify the semantic underpinnings of the conceptual knowledge-based visualization and exploration method, and (2) to demonstrate the feasibility of implementing that method.

4.1 The Temporal-Query Module

Starting a KNAVE session requires the user to select a domain ontology (e.g., "protocol-based care") via a direct link to the domain-ontology server. Once direct access to that ontology exists, the user can formulate a specific query for some entity, such as an abstract parameter or an external event (or *intervention*, as that particular knowledge role is relabeled by the display customization in a medical domain), by browsing the respective ontology. For example, using the static interface, the user can ask "show me all periods of *bone-marrow toxicity* of *any grade* in the *past 80 days,* in the context of *therapy by a Prednisone-Azathioprine (PAZ) protocol*" (a guideline for treatment of chronic graft-versus-host disease, a complication of bone-marrow transplantation) (Figure 3).



Start a Session

Patient: Peter Pumpkinhead

Time Unit: Days

Search for Context During

● Last 80 days

○ From Day 0 To Day 80

Context: PAZ

Parameter: Bone Marrow Toxicity

Parameter Value

● Any

○ Grade 0 to Grade 3

Figure 3: The temporal-abstraction query interface of the KNAVE system in a medical domain.

A temporal query defines the *parameter* type (e.g., bone-marrow toxicity), type of *abstraction* (e.g., STATE), parameter *value* restriction, if desired (e.g., any value), the *time* span (e.g., past 80 days), and the *context* (e.g., therapy by the PAZ protocol) (see Figure 3).

The display of the resultant temporal-abstraction intervals over time, which are returned by the TSA server, is quite straightforward, as will be seen in the next section, when we discuss the exploration operators. We have chosen simple linear representations of temporal intervals, as well as several standard visualizations of statistical descriptions (scattergrams, pie charts, distribution histograms, etc) when called for. We based the design on feedback from users in several different domains, mostly medical ones. Users often require more than piece of information at the same time (e.g., both the answer to a query regarding certain abstractions and the answer to a query regarding certain contemporaneous external events). Results appear in separate windows or panels, but temporal alignment among different windows and/or panels is the default.

4.2 The Dynamic Knowledge-Based Semantic-Exploration Operators

Users benefit greatly from direct interaction with the data and its temporal abstractions. We have noticed this benefit in previous experiments with automated temporal abstraction. The need for interaction was anticipated in previous medical-record–summarization programs, such as Downs' program (Downs 1987) and de Zegher-Geets' IDEFIX program (de Zegher-Geets, 1987). Both of these systems generated graphic displays of time-oriented clinical data at several levels of abstraction, and allowed a limited amount of interactive manipulation.

To exploit the full power of the temporal-abstraction mediator module (Tzolkin), and to support a domain-specific meaningful dialog, users need to be able to tap into the full knowledge contained within the domain model while interacting with the system. Once initial queries are answered, typical exploratory queries include questions such as "Show me other siblings classes at the same level of abstraction" (e.g., other hematological abstractions); "Show me the same data and resultant abstractions in a more specific context" (e.g., when a certain drug was given as part of the guideline); "What is the distribution of some raw or abstract data type (e.g., bone-marrow toxicity) during some period?" and "What if the data were a bit different: How would it affect the resultant patterns?" etc.

Thus, we have implemented a module that performs **knowledge-based exploration** through the visualized data and its temporal abstractions. This module is the core of the KNAVE architecture. We enable users to change the focus of the visualization by exploiting the domain knowledge that has been acquired from domain experts to support the creation of the temporal abstractions. In particular, we are using semantic links in the domain's temporal-abstraction ontology of parameters, interpretation contexts, and external events.

We have implemented six main semantic-exploration operators, corresponding to analog relations in the temporal-abstraction ontology (see Figures 4 and 5):

1. generalization/specialization of the parameter or pattern (e.g., from the white blood-cell count parameter to the class of hematological parameters and vice versa) using the IS-A semantic links among parameters;
2. functional dependency among parameters or patterns (e.g., from a bone-marrow toxicity abstraction in the PAZ context to the platelet-state and granulocyte-state abstractions defining it, and eventually, to the platelet-count and granulocyte-count parameter values defining these abstractions) using the ABSTRACTED-FROM or COMPOSED-FROM semantic links (Figure 5);
3. generalization/specialization of the interpretation context (e.g., from the preprandial (before meal) to the prebreakfast context in diabetes therapy) using the IS-A semantic links among interpretation contexts;
4. relations among contexts (e.g., from the context induced by the PAZ-therapy event to the context induced by a specific phase of that PAZ event, not necessarily contemporaneously) moving along the SUBCONTEXT relation;
5. generalization/specialization of the external event (e.g., from the insulin-administration event to regular-insulin administration) using the IS-A semantic links among events;
6. relations among events (e.g., from the PAZ-therapy event to a specific phase of the PAZ protocol in which a particular drug had been administered) moving along the SUBPART relation among events.

The knowledge-based semantic-exploration operators are demonstrated in Figure 4 as a set of seven hierarchical browser trees. Each tree focuses on one semantic relation, and usually also on its inverse. However, capitalizing on the initial feedback from our domain experts, we have split the dependency relation into the abstracted-from (derived from) and its inverse, abstracted-into (supports) relations to facilitate browsing. (This relation is a many-to-many relation, and thus cannot be depicted as a tree.) All browser trees are linked to the domain's temporal-abstraction ontology and enable dynamic exploration of that ontology.

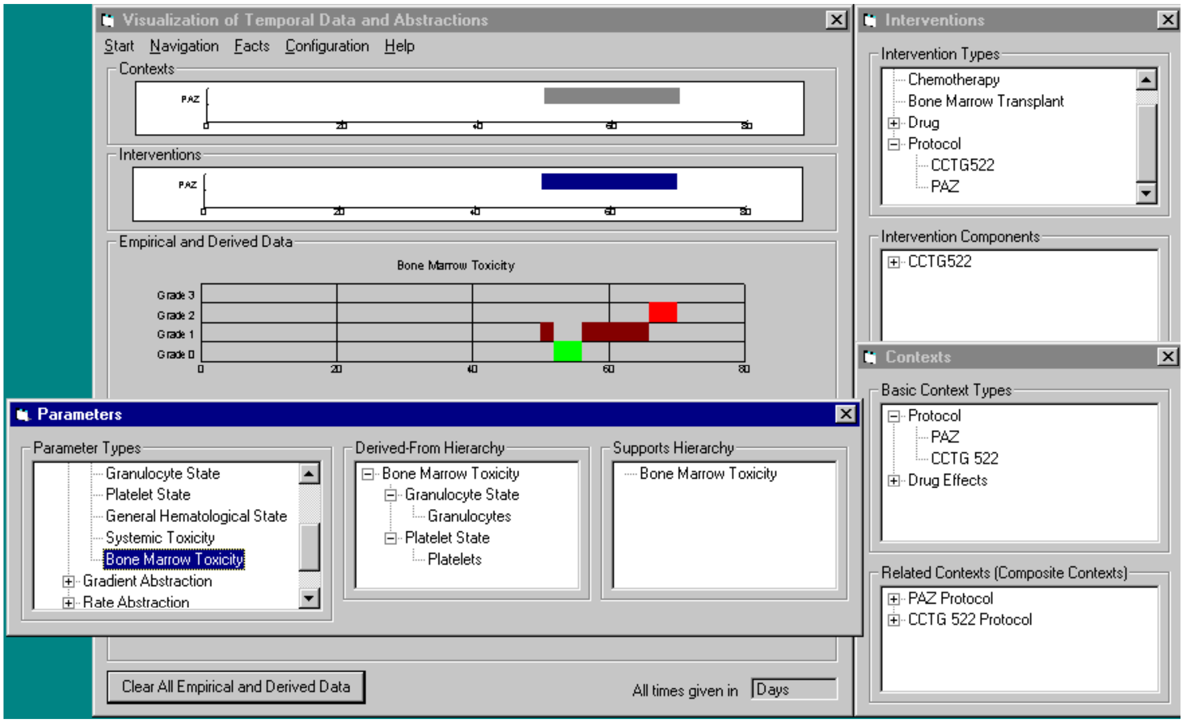


Figure 4: The interface to the dynamic knowledge-based semantic-exploration operators in KNAVE. Contexts and events ("interventions" in medical domains) are shown at the top. The results of the initial temporal query for bone-marrow toxicities displayed in Figure 3 are shown in the main panel. The seven browsers enable semantic explorations of different relations in the domain's temporal-abstraction ontology. The ABSTRACTED-FROM many-to-many relation had been split into the DERIVED-FROM and SUPPORTS relations to facilitate its exploration.

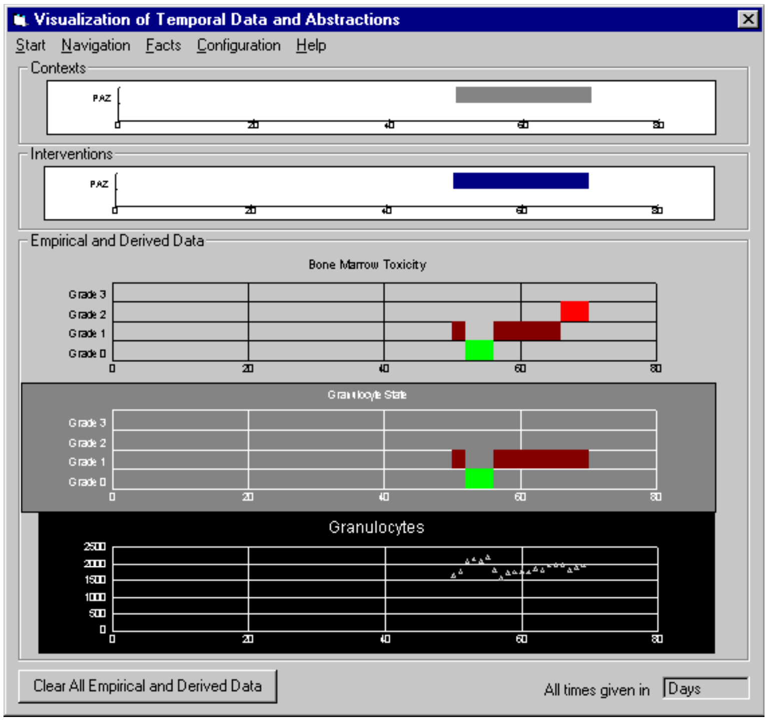


Figure 5: The result of an ABSTRACTED-FROM exploration query, given a bone-marrow-toxicity abstraction. The user has explored the bone-marrow-toxicity abstraction, moving along the ABSTRACTED-FROM relation to the granulocyte-state abstraction, one of the parameters from which it is derived. The user continued to explore the ABSTRACTED-FROM relation, arriving at the level of the granulocytes parameter, the raw data from which the granulocyte-state abstract parameter is derived.

Figure 5 demonstrates the results of a motion along the ABSTRACTED-FROM (*derived from*, as the medical-domain customization prescribed in this case) semantic link, following an initial query that displayed a bone-marrow-toxicity abstraction interval. The user chose to perform a "depth-first" exploration into the granulocyte-state supporting abstraction, and into the granulocyte-count raw data supporting that abstraction. The user could also display the parameters supporting the bone-marrow-toxicity abstraction in a "breadth-first" manner, displaying both the granulocyte-state and platelet-state abstractions from which the bone-marrow-toxicity abstraction is derived (see Figure 4).

4.2.1 Additional Temporal-Syntactic and Temporal-Semantic Exploration Operators

We have added several additional operators that facilitate browsing and exploration; several of these are purely *syntactic* (context-free, depending only on the data), and several are *semantic*, in the sense that they rely on access to the domain ontology and its semantics.

For instance, to facilitate browsing of data and abstractions we have added a *syntactic temporal-zoom (magnification) operator*. Thus, selecting this operator and one panel or a part of it in a visualization window (e.g., just platelet counts) expands the display of the data within the selected time and data period to fill the whole screen, magnifying the display and enabling the user to inspect data or abstractions shown during that time interval more closely. We have also added operators such as *temporal overlay* of different types of data along the same timeline (Cousins and Kahn, 1991), *horizontal scrolling* along the temporal dimension, and *vertical scrollin*g along the abstraction dimension. We have also added standard descriptive statistical operators that work at any level of abstraction, such as showing the distribution of the granulocyte-state abstraction values during a selected period.

4.2.1.1 The temporal-semantic-zoom operator

To facilitate summarization and exploration of large amounts of data over significant time periods, we are enabling users to change dynamically the *temporal granularity* of the interactive interpretation and visualization process. Often, users need to examine data at different temporal resolution levels (e.g., minutes, days, or years). Furthermore, certain levels of temporal granularity (e.g., weeks or days) are not meaningful for visualization of certain types of data (e.g., heart rates in the intensive-care–unit context need to be shown at the level of seconds or minutes, in which these data are typically acquired). Thus, each type of data typically has certain *ranges of temporal granularit*y within which visualization is meaningful and beyond which the data should be visualized differently, by using a domain-specific *aggregation operator* (e.g., by using descriptive statistics such as minimum, maximum, mean, standard deviation, distribution, etc., or moving to a higher abstraction of the parameter).

We enable users to change on the fly the temporal granularity level of the interface (i.e., the relevant resolution of the presentation and interpretation process, as opposed to the syntactic temporal zoom, which involves only simple graphical magnification of the data display but adds no details or information). The knowledge of (1) which temporal-granularity ranges are meaningful for each parameter in each context, and (2) what aggregation operators should be used automatically when the granularity chosen is out of that range, is acquired and added to the domain model as part of the visualization knowledge. This knowledge is acquired from the domain experts in the process of acquiring the temporal-abstraction knowledge. The need for explicit representation of relevant temporal-granularity ranges in the domain's model was noted by Cousins and Kahn (1991), although their main goal was the development of a syntactic model for visualization of time-oriented clinical data, rather than exploration of the data abstractions using the domain's knowledge.

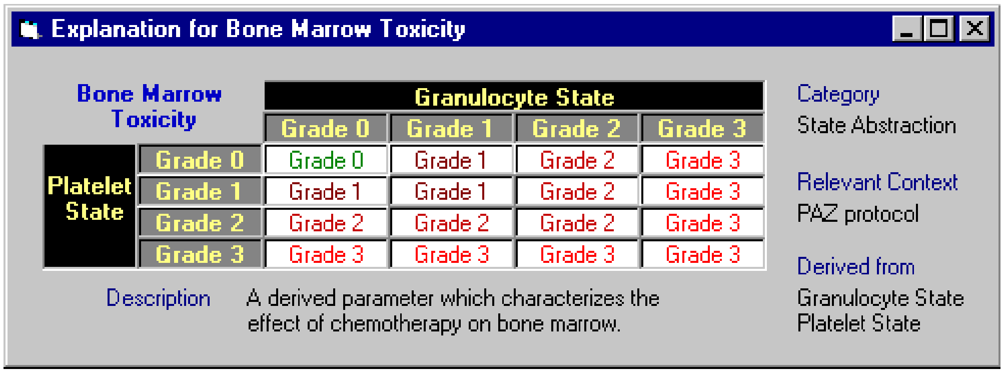
When domain-specific visualization knowledge is lacking, the temporal-semantic-zoom operation uses display heuristics of the type that have been explored in previous work on automated design of user interfaces (Gnanamgari, 1981; Arens et al., 1994). The heuristics use only *syntactic* properties of the data; for instance, quantitative, numeric data are usually summarized using mean and standard deviation; qualitative, symbolic data are usually be summarized using distribution histograms or pie charts (if all allowed values are known) at the appropriate level of temporal granularity (e.g., one histogram per month, if month is the minimal temporal unit).

4.2.2 Interactive Knowledge-Based Explanation

Users often require meaningful explanations for either data interpretations or action recommendations, such as when a physician examines the visual results of a temporal query to a patient's record.

Using the tight link between the KNAVE core visualization client and the domain ontology server (see Figure 2), we provide users with various types of context-sensitive explanations. For instance, we may need to answer a query such as "*why* is this interval characterized as bone-marrow toxicity grade 3?" (in addition to the query "from *which* data is it abstracted?" which can be answered by semantic exploration along the ABSTRACTED-FROM link).

To answer that query, we retrieve from the domain's ontology the knowledge (in this case, a classification function that is represented in the knowledge-acquisition tool and the temporal-abstraction ontology of the domain as a table) that defines the abstraction of bone-marrow toxicity grades from the toxicity levels of platelets and granulocytes in a context-sensitive fashion (e.g., specific to the PAZ-therapy context in which the abstraction was created) (Figure 6).



Explanation for Bone Marrow Toxicity

Bone Marrow Toxicity		Granulocyte State			
		Grade 0	Grade 1	Grade 2	Grade 3
Platelet State	Grade 0	Grade 0	Grade 1	Grade 2	Grade 3
	Grade 1	Grade 1	Grade 1	Grade 2	Grade 3
	Grade 2	Grade 2	Grade 2	Grade 2	Grade 3
	Grade 3	Grade 3	Grade 3	Grade 3	Grade 3

Category
State Abstraction

Relevant Context
PAZ protocol

Derived from
Granulocyte State
Platelet State

Description A derived parameter which characterizes the effect of chemotherapy on bone marrow.

Figure 6: Providing an explanation to a query about the bone-marrow-toxicity abstractions shown in Figure 4 by displaying a classification (functional-dependency) table from the temporal-abstraction knowledge base.

4.2.3 Special Interactive Operators: Dynamic Manipulation of Data and Interpretation Contexts

Apart from the exploration operators described in this section, which involve the user only in the sense of applying predefined exploration operators and which we had implemented in straightforward fashion, there are two types of interactive exploration operations which involve direct manipulation, by the user, of the explored contents. These special operators pose significant demands on the conceptual and practical communication capabilities within any knowledge-based visualization and exploration framework, and in particular between the three KNAVE modules. Based on the insights we have gained from development of the KNAVE prototype, we are adding these two operator types to the current implemented prototype.

4.2.3.1 Controlling the overall goal of the abstraction process

Often, the overall context for the abstraction process can be induced automatically from the database record and the domain's temporal-abstraction ontology. For instance, therapy by the PAZ protocol induces a PAZ-therapy interpretation context, contemporaneous with the intervention interval; within that time interval, clinical data (e.g., hematological parameters) will be interpreted by the RÉSUMÉ system in a manner specific to that context. However, often the appropriate context for the abstraction process cannot be derived automatically from the data. For instance, when a physician examines the record of a patient who is known to have, among others problems, diabetes, she would like to visualize a summary of the patient's data during the past four weeks, *from the point of view of supporting a diabetes therapy goal*. She would not want to interpret *all* types of data in *all* potentially useful contexts, and certainly not throughout the patient's history. This goal, however, is a conceptual one, existing only in the mind of the care provider. It needs to be stated explicitly, so that it can be part of the database (at least temporarily) and thus can induce the appropriate interpretation context(s) (through the context-forming mechanism).

Thus, we are currently adding an interactive dialog that enables users, during the visualization session, to dynamically add to the interpretation process *abstraction goals* (Shahar, 1997), whose main object is to set the overall context(s) and relevant time span(s). The overall context enables the context-sensitive mechanisms of the RÉSUMÉ system to create meaningful abstractions specific to that context. Such a goal-oriented mode of performing the abstraction process is both necessary, efficient, and supports a useful dialog with the end user.

4.2.3.2 Interactive hypothetical queries

Users often wonder if another specific datum would have changed the overall view of some segment of the data, when either added or removed. Thus, we are currently adding the capability to enable at runtime addition and retraction of data to and from the KNAVE dynamic memory. Thus, we enable the user to ask *hypothetical (“what-if”) queries* by hypothetically asserting or retracting data (e.g., “what if the platelet count during the previous visit was in fact 50,000 and not 80,00?”) and by visualizing the resulting abstractions (e.g., the bone-marrow-toxicity grade might change to Grade III during the time interval corresponding to that visit, while certain abstraction might disappear). Both the abstraction capability and the what-if queries are already supported by the RÉSUMÉ system and its specialized truth-maintenance system, which maintains logical dependencies among data and their abstractions (Shahar and Musen, 1996). The RÉSUMÉ truth-maintenance system supports the nonmonotonic nature of temporal abstraction, whose conclusions are always potentially defeasible by additional data.

4.3 Explicit Representation of Domain-Specific And Individual User Models

The KNAVE system should fit its interface terms and semantics (beyond the temporal-abstraction ontology) to the needs of each *domain*, and should accommodate the needs of different *user groups* and even *individual user*s within each domain. Thus, we enable a customization of the interface terms and behavior for each domain, including the listing of the relevant groups of users in each domain (e.g., physicians, nurses, etc., in the case of medical domains) and their default preferences, and by also enabling customization by individual users in that domain. These customizations are part of the domain's (visualization) model. Thus, in the case of medical domains, we have added defaults specifying how to present data to various groups of care providers such as nurses, interns, attending physicians, social workers, etc. Each individual user can further edit her group profile to create her own "user profile" and save it for future use. Examples include preferences such as whether to show just top level abstractions or all intermediate ones when returning the top-level abstraction in response to a temporal query (and similarly, whether to show just overall events or also their parts); over what span of time to start the initial query; what classes of data should be retrieved by the initial query (e.g., a diabetes-therapy expert might want to always start a visualization session by querying for the presence of certain periodic abstractions); and what types of abstractions are, in general, relevant to the user (e.g., an intensive care nurse might be interested, as a default, only in rate and gradient abstractions of certain classes of parameters such as blood gases and vital signs).

The user interface module is being designed using a *model-based approach*. This means that in addition to the temporal-abstraction ontology, which is used to interpret data, there is also a domain model for the display of the data and its abstractions. Thus, the *domain display model* includes a set of arguments for modifying the layout, order of presentation, and customization of the behavior of semantic-exploration operators. Examples of using the default model for the set of medical domains can be found within all screen displays shown in this paper. For instance, note the substitution of widget labels to provide domain-specific naming schemes. In the medical domain, concepts such as "primitive parameters", "abstract parameters", and "events," which originate from the ontology of the knowledge-based temporal-abstraction problem-solving method, were *mapped* (Gennari et al., 1994), respectively, to the terms "empirical data", "derived parameters", and "interventions" which are the default mappings for a medical domain.

The domain model thus specifies a default *user profile*. This default profile can then be customized to fit the needs of particular user groups (such as nurses) within the domain. Group profiles can be further specialized to accommodate individual users.

4.4 Implications of the KNAVE Architecture for the Temporal Abstraction Framework

We have found it necessary to make several enhancements to the implementation of the knowledge-based temporal-abstraction method in the RÉSUMÉ system and to the temporal-abstraction knowledge-acquisition tool to support the KNAVE visualization and exploration requirements. Most of these enhancements or their analogs would be expected even if the implementation details were quite different.

1. We have enhanced the RÉSUMÉ system to enable creation of more complex abstraction patterns, and in particular, patterns comprising both *temporal* and *statistical* aspects. For example, a typical need in the diabetes-therapy domain is to visualize an *increasing* gradient of the weekly *variance* of the blood-glucose after breakfasts over the past 6 weeks. Such an abstraction, however, requires the knowledge to create the correct interpretation context. In this example, it is a prospective postprandial (after meals) context of the nonconvex type (i.e., composed of temporally disjoint intervals), specialized

to the post-breakfast context, that is *induced* by the existence of breakfast events in the database (Shahar, 1998). Then, a purely statistical abstraction (variance) needs to be computed over each relevant time granule (here, a week). Finally, the resulting (gradient-type) abstraction should be able to serve as input to a higher-level pattern.

2. We have enhanced the graphical temporal-abstraction knowledge-acquisition tool so as to acquire more complex linear temporal patterns involving temporal-distance, temporal-relations, and parameter-value constraints. These patterns are first-class entities in the domain's ontology. Linear patterns have COMPOSED-OF relations to their components. Components are abstractions or patterns that have internal, *local* constraints on start time, stop time, duration, and value of the abstraction. *Global* qualitative and quantitative temporal and value constraints exist between components. Accordingly, we have enhanced the RÉSUMÉ temporal-pattern matching mechanism to detect and create these patterns at runtime.
3. We have added the ability to represent and detect *periodic temporal patterns* and, in general, any *repeating pattern* of a certain class. We represent periodic patterns as a repeating linear pattern with certain constraints such as cardinality and temporal-gap (e.g., each day, or 2 to 4 times a week) (Chakravarty and Shahar, 1999). Periodic patterns can be specified in the graphical knowledge-acquisition tool, abstracted by the RÉSUMÉ system, and visualized as regular interval-based abstractions within the KNAVE system.
4. We have enhanced the knowledge-acquisition tool to enable acquisition of necessary *temporal-visualization knowledge*. Examples include the relevant temporal-granularity ranges that should be used for visualization of each parameter, the default abstraction or statistical functions that should be used automatically when the visualization granularity is

outside of the scope of the relevant temporal granularity, and domain-specific interface customizations.

5. Certain changes needed to be carried out with respect to the fashion in which any temporal mediator, and in particular the Tzolkin control module, coordinates the actions of the RÉSUMÉ temporal-abstraction module and the Chronus temporal-maintenance module. For instance, intermediate abstractions created by the RÉSUMÉ system need to be returned as part of the output, for exploration purposes. Currently, we are accessing the RÉSUMÉ conclusions directly. In the future, queries to the Tzolkin module will require more expressive semantics.

5 PRELIMINARY EVALUATION OF THE KNAVE FRAMEWORK

As emphasized in the introduction, this paper focuses on the knowledge-based visualization and exploration methodology and its semantics, and the architecture implied by that method. However, to demonstrate the feasibility of the methodology before we proceeded with a large-scale implementation, we have performed a preliminary developmental assessment of a prototype version of the KNAVE core module (the visualization and exploration components) in the domain of protocol-based care in clinical medicine (Cheng et al., 1997). The preliminary evaluation can be viewed as a pilot study to test our hypothesis that a task-specific knowledge-driven visualization and exploration system is both feasible and useful. This evaluation also provided preliminary feedback from potential users in various different medical domains.

To focus on a preliminary evaluation of the visualization and exploration module in a medical domain, we ignored the communication links. Both the time-oriented patient data and its abstractions (generated by the RÉSUMÉ system) and the domain temporal-abstraction ontology were represented as simple files on the same workstation. A single patient record was created from two patient files (one presenting a patient after bone-marrow transplantation and treated by an experimental protocol to counter the complication of graft-versus–host disease, and the other presenting a patient was treated by an experimental protocol for AIDS therapy), since we wanted to demonstrate several different interpretation contexts. The patient was presented to the users as an AIDS patient who underwent a bone-marrow transplantation for therapy of oncological complications. The prototype, whose screen displays were shown in Figures 3 to 6, included a subset of the static and dynamic, semantic and syntactic operators that we are implementing in the reengineered system. Seven users with varying medical and computer-use backgrounds were given a 10-minute explanation of the knowledge-based visualization and exploration framework; the users were not otherwise familiar with the KNAVE system. The users were then requested to answer within 20 minutes three complex temporal queries about the particular set of data we used, using only the new interface and a brief introduction to the exploration interface. The queries were realistic and came from the domain of guideline-based care in oncology and AIDS therapy. The queries were designed to test multiple aspects of the KNAVE architecture, while being typical of tasks that clinicians are required to perform as part of treating patients using complex clinical guidelines, as is common in the domains used here. The first query inquired whether there was ever any episode of at least one week of bone-marrow toxicity grade II or more, in the context of the prednsone/azathioprine (PAZ) protocol for therapy of graft-versus-host disease, and if so, on which data that conclusion was based. The task involved query, visualization, and some explanatory exploration (see Figures 4 and 5). The second query introduced the concept of different interpretation contexts, and asked about whether certain values of the hematological-state abstraction in the CCTG-522 AIDS-therapy experimental protocol can be found during any period. The third task forced the users to use the statistical-abstraction and visualization capabilities as well, since it focused on finding the distribution of the hemoglobin-state abstraction values in the CCTG-522 context over the past 80 days.

The results, although obviously limited by the small number of users and the informal questions they were asked following the experiments, were quite encouraging with respect to the users' subjective enthusiasm towards the framework's potential and the objective capability of the prototype to provide visual answers to the various queries that the users tried to answer. All users completed the three tasks (that is, answered the three complex temporal queries) within 20 to 180 seconds overall; six of seven users performed the three tasks within less than 90 seconds. The users emphasized the importance of being able to get immediate explanations to the visualized abstractions, and seemed to like especially the combination of temporal and statistical abstractions.

One of the lessons we learned, which we will emphasize in the final implementation, was the importance of *redundancy*. Somewhat surprisingly, different users found answers to the same queries using several (up to four) different paths to get to the same visualized set of abstractions (e.g., through the semantic-exploration interface, by going back to the initial temporal-query interface, by using the navigation drop menu, or by using a short-cut mouse right-button function).

6 DISCUSSION AND COMPARISON WITH RELATED WORK

It is difficult to compare the results of the preliminary evaluation to any current automated interface, since performance of the task requested of the users requires having both the ability to abstract clinical data over time into multiple levels of concepts, and the access to a suitable domain-specific knowledge base (such as the temporal-abstraction ontology). We know of no such architecture other than ours. One might compare the performance of the users of KNAVE to the manual one we witnessed while evaluating the RÉSUMÉ system in the diabetes domain (Shahar and Musen, 1996). In that domain, manual abstraction of somewhat similar queries required up to 10 minutes per query by either of the two experts (one of whom participated as a user in the preliminary KNAVE evaluation). We also plan to compare KNAVE to an online interface, such as an electronic spreadsheet, that has charting capabilities but no explicit, application-independent, knowledge representation.

Visualization of information in general and of large amounts of time-oriented data in particular is essential for effective decision making. Much effort had been put in the past into creation of effective visualizations for information; an excellent example is the classic series of books by Edward Tufte on methods to display information (Tufte 1983, 1990, 1997).

Previous work had typically focused on exploring separately three different subtasks of the problem we are tackling: *temporal abstraction*, *information visualization*, and *knowledge acquisition*. It is not a coincidence that these three subtasks are represented explicitly in the KNAVE architecture as three distributed modules.

Several approaches have been applied to the task of *abstraction of time-oriented data into higher-level concepts*, especially in medical domains, in which both large amounts of data and considerable knowledge are available (Fagan, 1980; Downs et al., 1986; De Zegher-Geets, 1987; Kohane, 1987; Russ, 1989; Kahn, 1991; Larizza et al., 1992; Haimovitz and Kohane, 1993). None of these approaches, however, emphasized the need for a formal representation that facilitates acquisition, maintenance, sharing, and reuse of the required temporal-abstraction knowledge; this emphasis is the focus of our previous and current research. Furthermore, previous temporal-abstraction approaches were typically not geared for use in a runtime system for visualization of and exploration through the domain-specific abstractions, and would not support a domain-independent interface.

Researchers in the areas of *presentation and display* techniques in general (Tufte 1983, 1990), *visualization* of *clinical time-oriented data* (Cousins and Kahn, 1991; Powsner and Tufte, 1994; Plaisant et al., 1996), and *human-computer interfaces*, in particular *information visualization* (Rao et al., 1992; Roth et al., 1994; Shneiderman, 1994; Arens et al., 1994; Zhou and Feiner, 1996; Koljejchick et al., 1997; Rohrer and Swing, 1997; Wright, 1997; Becker, 1997) have developed useful visualization techniques for static presentation of *raw* time-oriented quantitative data and for browsing information. These techniques use various statistical and graphical methodologies, such as scattergrams, pie charts, bar charts, three-dimensional representations (Mackinlay et al., 1994; Carpendale et al., 1997) and their derivative techniques. These display methods, however, typically do not focus on visualization of domain-specific *temporal abstractions* (as opposed to raw data) and on the issue of *dynamic exploration*, using a domain-independent method, through *multiple levels* of these abstractions using domain-specific knowledge. The reason for that omission is that such capabilities require formal, domain-independent representations of the domain-specific temporal-abstraction knowledge, considerable effort in modeling the visualized domain, and the availability of computational mechanisms for creation of the relevant abstractions.

The past decade has witnessed considerable advances in semiautomated methods for *knowledge acquisition* and knowledge representation, based on approaches that operate at the knowledge level (Newel, 1982) and that assume task-specific but domain-independent problem-solving methods (Clancey, 1985; Chandrasekaran, 1986; McDermott, 1988; Musen, 1989; Weilinga et al., 1992, Genesereth and Fikes, 1992; Gruber, 1993; Eriksson et al., 1995) which often succeed in alleviation of the knowledge-acquisition bottleneck. However, these methods are not often associated with runtime end-user applications; instead, they focus on use by knowledge engineers and domain experts. In the case of visualization tools in particular, existing frameworks are typically intended to assist in visualization and exploration of *knowledge* by *knowledge engineers*, rather then exploration of *data* by *domain end users* (Jones, 1988; Eisenstadt, 1990).

There also has been considerable useful work on model-based design of graphic user interfaces (Puerta, 1997). Such work considers domain ontologies and properties in designing a specific interface for domain applications. There has also been very interesting work on automated, task-analytic design of user interfaces, which considers domain properties as well as the desiderata of different tasks in the application domain (Casner, 1991). Our approach, however, is quite different. We have designed a visualization and exploration interface that is *not* specific to any application *domain* (apart from certain domain- and user-specific customizations), but that is specific only to a well defined set of *tasks* involving exploration of time-oriented data and its abstractions. Furthermore, the approach is specific to a particular *problem-solving method* (the knowledge-based temporal-abstraction method) and its domain-independent *method ontology*. Thus, we are *not* attempting to design automatically the user interface based on analysis of either the task (e.g., temporal abstraction) or the application domain (e.g., oncology). Instead, the temporal-exploration task has been analyzed in our methodology ahead of time, and the domain's ontology is mapped to the terms of the formal ontology of the knowledge-based visualization and exploration method (an extension of the ontology of the knowledge-based temporal-abstraction method). The semantics of that problem-solving method also determine the meaning of most of the exploration operators. Therefore, the user interface is essentially similar (except for group and individual preferences) in all domains in which the temporal-exploration task needs to be solved.

The drawback of our task-specific approach is therefore a trade off: We do not and cannot design new interfaces on the fly for specific application domains, or use any domain-specific (rather than task-specific) graphical paradigms. Examples might include drawing pediatric growth charts as nonlinear curves, as they are often used by pediatricians to record the raw data, or using helices and other domain-specific conventions to represent biomolecular information such as DNA coils. However, once data are abstracted into intervals and presented in the context of the temporal-abstraction (or any other linear-abstraction) task, even when the raw data use domain-specific graphics, they might be readily understood by domain experts, as our experience has shown in the domain of monitoring of children's growth (Kuilboer et al., 1993).

We can summarize and highlight the main features of our methodology in somewhat more detail, using terminology that has been found useful in the knowledge-acquisition research area. We have presented in this paper a conceptual and computational knowledge-based framework for interactive visualization and exploration of time-oriented data and their multiple levels of temporal abstractions that is quite different from previous work. In this framework, the semantics of the querying, visualization, exploration, and explanation operators are defined by the domain-independent ontology of the knowledge-based temporal-abstraction problem-solving method. The semantic-exploration operators allow the runtime user to explore visually the domain-specific temporal-abstraction (sub) ontologies, thereby leading to reciprocal visual exploration through the multiple levels of temporal abstractions of the particular time-oriented database that is queried. Thus, although the KNAVE interface is essentially the same (apart from certain domain- and user-dependent customizations) in every domain, and the semantics of the syntactic and semantic exploration operators are identical, the resultant browsing process is specialized to the domain's parameter, event, context, and abstraction-goal (sub) ontologies, which the knowledge-based temporal-abstraction ontology comprises.

The visualization and exploration modules of the KNAVE architecture and the inference structure underlying them can therefore be viewed as part of a new problem-solving method, specific to the task of visualization of and exploration through time-oriented data and its temporal abstractions. The ontology of this *knowledge-based visualization and exploration method* is a superset of the ontology of the knowledge-based temporal-abstraction problem-solving method, augmenting it with additional knowledge such as new classes (e.g., statistical abstractions) and properties (e.g., visualization knowledge, such as what temporal resolution range is relevant for the visualization of each parameter) that are specific to the interactive-visualization task. The domain-specific temporal-abstraction knowledge acquired from domain experts is thus *reused* by a different problem-solving method, which solves a different task (visualization and exploration, rather than temporal abstraction) and is *shared* by different applications within the same domain. The new method also reuses the computational mechanisms of the knowledge-based temporal-abstraction problem-solving method and augments them by the visualization and exploration mechanisms.

Note that, using the concept of *syntactic* and *semantic* (knowledge-based) exploration operators, we can more accurately define, conceptually and practically, several categories of *composite exploration operators*, each of which can be defined as a specific combination of semantic-exploration and syntactic temporal-exploration operations. For instance, one variation of what has often been called "drilling down the data" in certain information-visualization systems, could be more formally defined as a composite exploration operator that consists of a syntactic temporal zoom (simple magnification) combined with moving along an ABSTRACTED-FROM semantic link. Other types of "zooming," differing significantly from each other, can likewise be easily defined.

The applicability of the RÉSUMÉ system to strikingly different time-oriented domains such as guideline-based medical care, monitoring of children's growth, therapy of diabetes, and traffic control suggests that the new knowledge-based visualization methodology will be widely applicable as well. Furthermore, the abstraction experiments within the traffic-control domain (Shahar and Molina, 1998) have demonstrated that the knowledge-based temporal-abstraction methodology is useful for acquisition and maintenance of other types of linear-abstraction knowledge, such as spatial-abstraction knowledge. These experiments suggest similar applicability of the knowledge-based visualization methodology to domains that employ linear distance measures and to the semiautomated (interactive) support of tasks such as monitoring, planning, explanation, and data mining. The evaluation of the final reengineered KNAVE system in several different domains will provide further insights into these intriguing hypotheses.

7 FINAL NOTES: CURRENT AND FUTURE WORK

Based on the conceptual and practical insights we have gained from our work on the KNAVE architecture, its preliminary prototype, and its developmental evaluation, we are currently in the process of reimplementing our visualization and exploration module prototype (originally implemented in Visual Basic) using the Java programming language. The architecture, however, has not changed, neither did the semantics of the visualization and exploration operators.

The KNAVE architecture comprises three main conceptual components (see Figure 2). Thus, communication links exist between the knowledge-based visualization and exploration module, the domain ontology server, and the temporal-database mediator, with an integrated view in mind (Wiederhold et al., 1986). The communication protocols support both the static and dynamic data (raw input), information (abstracted conclusions) and knowledge (from the domain's temporal-abstraction ontology) requirements of the KNAVE system. The communication links to both the ontology server and the database have been implemented using the **common object request broker architecture** (**CORBA**) standard. The CORBA standard is a specification standard, increasingly adapted by the software industry, that describes how software components can interoperate across networks, languages, and platforms. The use of CORBA makes it straightforward for the components that we are using to be developed in various programming languages, to reside on different computers, and to communicate with one another over the Internet. The CORBA standard provides an standard interface that allows arbitrary software modules to pass data to the knowledge-based components, and to receive the resulting inferences in a transparent fashion.

In the medical domain, users select the patient record to be explored and the relevant domain temporal-abstraction knowledge base to be used to interpret that record (e.g., protocol-based care or diabetes). Selection of the knowledge base enables runtime creation of a domain-specific temporal-abstraction browser (a somewhat compact version of the seven browsers seen in Figure 4) and thus formulation of a domain-specific query. The results are displayed in the visualization window. The most common exploration operators (e.g., drilling down the abstraction hierarchy) are part of the window, so users need only return to the full domain browser for significantly new queries. New queries (or explorations) add new panels to the same window, unless the user elects to open a new window.

Given the importance of feedback, we are performing developmental assessments and evaluations from the very beginning of the final implementation of the KNAVE architecture, using our collaborators in several medical domains and clinical data sets, several of which we already have, and several of which our collaborators supply as necessary (e.g., a large data set in the bone-marrow transplantation domain).

We are also evaluating the overall KNAVE architecture within EON, an architecture for guideline-based medical care (Musen et al., 1996). EON is being applied to domains such as protocol-based care for AIDS patients and oncology.

Our work in the traffic-control domain (Shahar and Molina, 1998) potentially provides a significantly different domain for assessing the effectiveness of our domain-independent methodology. It is our hypothesis that our methodology is useful in most domains in which data are captured over a linear distance measure (such as uni-dimensional space along each highway, and the evolution of the spatial abstractions over time), and in which there exist several levels of abstraction of the data.

Once the final reengineering of KNAVE is done, we will perform a larger and more formal evaluation of the KNAVE framework, including more users, qualitative and quantitative questionnaires, “thinking aloud” experiments, inclusion of well-defined target user behaviors, and application of standard methods for interface usability assessments (Gould, 1988). We will try to assess the benefits of the new methodology versus browsing the data on paper or in a standard online format such as an electronic spreadsheet.

Our ultimate vision is to enable exploration of any distributed time-oriented database (e.g., on the internet), using an appropriate existing temporal-abstraction ontology (e.g., a knowledge base located somewhere else on the World Wide Web), from any visualization and exploration workstation. Needless to say, this vision requires acquisition of significant amounts of temporal-abstraction knowledge. We have taken the first steps towards formalization and acquisition of that knowledge in our previous work, and are now utilizing it to enable exploration of the resultant temporal abstractions.

Acknowledgements

This work has been supported by grants LM06245 from the National Library of Medicine and IRI-9528444 from the National Science Foundation. Drs. Daniel Stites, Lawrence Basso, Darrel Wilson, and Herbert Kaizer provided input regarding the design of the preliminary interfaces for both the graphical knowledge-acquisition tool and the KNAVE visualization and exploration module, and many users tested our prototype in the evaluation study. Dr. Angel Puerta provided useful references and advice in the preliminary evaluation phase.

References

ALLEN, J.F. 1984. Towards a general theory of action and time. Artificial Intelligence **23**: 123–154 .

ARENS, Y., L. MILLER, S. C. SHAPIRO, and N. K. SONDHEIMER. 1994. Automatic construction of user-interface displays. *In* Proceedings of the Seventh National Conference on Artificial Intelligence, St. Paul, Minnesota, pp. 808–813.

BECKER, B. G. 1997. Using MineSet for knowledge discovery. IEEE Computer Graphics and Applications, **7/8**: 75–78.

BLUM, R.L. 1982. Discovery and representation of causal relationships from a large time-oriented clinical database: The RX project. In Lindberg, D.A. and Reichartz, P.L. eds, Lecture Notes in Medical Informatics, volume 19, New York: Springer-Verlag.

CARPENDALE, M. S. T., D. J. COWPERTHWAITE, and F. D. FRACCHIA. 1997. Extending distortion viewing from 2D to 3D. IEEE Computer Graphics and Applications, **7/8**: 42–51.

CASNER, S. M. 1991. A task analytic approach to the automated design of graphic presentations. ACM Transactions on Graphics **10**(2):, 111–151

CHAKRAVARTY, S., and Y. SHAHAR. 1999. A constraint-based specification of periodic patterns in time-oriented data. *In* Proceedings of the 1999 Sixth International Workshop on Temporal Representation and Reasoning Time-99, Orlando, Florida, pp. 29–40.

CHANDRASEKARAN, B. 1986. Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. IEEE Expert, **1**: 23–30.

CHENG, C., Y. SHAHAR, A. R. PUERTA, and D. P. STITES. 1997. Exploration and visualization of abstractions of time-oriented clinical data. Section on Medical Informatics Technical Report No. SMI-97-0688, Stanford University, CA.

CLANCEY, W. J. 1985. Heuristic Classification. Artificial Intelligence, **27**: 289–350.

COUSINS, S. B., and M. G. KAHN. 1991. The visual display of temporal information. Artificial Intelligence in Medicine, **3**: 341–357.

DAS, A. K., and M. A. MUSEN. 1994. A temporal query system for protocol-directed decision support. Methods of Information in Medicine, **33**(4): 358–370.

DAS, A. K., Y. SHAHAR, S. W. TU, and M. A. MUSEN. 1994. A temporal-abstraction mediator for protocol-based decision support. *In* Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC, pp. 320–324.

DE ZEGHER-GEETS, I. M. 1987. IDEFIX: Intelligent summarization of a time-oriented medical database. M.S. Dissertation, Medical Information Sciences Program, Stanford University

School of Medicine, June 1987. Knowledge Systems Laboratory Technical Report KSL–88-34, Department of Computer Science, Stanford University, Stanford, CA.

DOWNS, S. M., M. G. WALKER, and R. L. BLUM. 1986. Automated summarization of on-line medical records. *In* MEDINFO '86: Proceedings of the Fifth Conference on Medical Informatics, *Edited by* R. Salamon, B. Blum, and M. Jorgensen, North-Holland, Amsterdam, pp. 800–804.

EISENSTADT, M., J. DOMINGUE, T. RAJAN, and E. MOTTA. 1990. Visual knowledge engineering. IEEE Transactions on Software Engineering, **116**(10): 1164–1177.

ERIKSSON, H., Y. SHAHAR, S. W. TU, A. R. PUERTA, and M. A. MUSEN. 1995. Task modeling with reusable problem-solving methods. Artificial Intelligence, **79**(2): 293–326.

FAGAN, L. M. 1980. VM: Representing Time-Dependent Relations in a Medical Setting. Ph.D. dissertation, Department of Computer Science, Stanford University, Stanford, CA.

GENESERETH, M. R., and R. E. FIKES. 1992. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, Stanford, CA.

GENNARI, J. H., S. W. TU, T. E. ROTHENFLUH, and M. A. MUSEN. 1994. Mapping domains to methods in support of reuse. International Journal of Human-Computer Studies. **41**: 1994 399–424.

GIARRATANO, J., and G. RILEY. 1994. Expert Systems: Principles and Programming. Boston, MA: PWS Publishing Company.

GNANAMGARI, S. 1981. Information Presentation Through Default Displays. Ph.D. dissertation, University of Pennsylvania.

GOULD, J. D. 1988. How to design usable systems. *In* Handbook of Human-Computer Interaction. *Edited by* M. Helander, Elsevier Science Publishers, pp. 757–789.

GRUBER, T. G. 1993. A translation approach to portable ontology specification. Knowledge Acquisition, **5**: 117–243.

HAIMIOWITZ, I. J., and I. S. KOHANE. 1993. Automated trend detection with alternate temporal hypotheses. *In* Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, San Mateo: Morgan Kaufmann, pp. 146–151.

JONES, S. 1988. Graphical interfaces for knowledge engineering: An overview of relevant literature. Knowledge Engineering Review, **3**(3), 221–248.

KAHN, M. G., C. A. ABRAMS, S. B. COUSINS, J. C. BEARD, and M. E. FRISSE. 1990. Automated interpretation of diabetes patient data: Detecting temporal changes in insulin therapy. *In* Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care. *Edited by* R. A. Miller. Los Alamitos: IEEE Computer Society Press, pp. 569–573.

KAHN, M. G. 1991. Combining physiologic models and symbolic methods to interpret time-varying patient data. Methods of Information in Medicine, **30**: 167–178.

KOHANE, I. S. 1987. Temporal reasoning in medical expert systems. Technical Report 389, Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

KOLOJEJCHICK J., S. F. ROTH, and P. LUCAS. 1997. Information Applications and Tools in Visage. IEEE Computer Graphics and Applications, **7/8**: 32–41.

KUILBOER, M. M., Y. SHAHAR, D. M. WILSON, and M. A. MUSEN. 1993. Knowledge reuse: Temporal-abstraction mechanisms for the assessment of children's growth. *In* Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medicine, Washington, DC, pp. 449–453.

KUMAR, H.P., C. PLAISANT, and B. SHNEIDERMAN. 1997. Browsing hierarchical data with multi-level dynamic queries and pruning. International Journal of Human-Computer Studies 46(1) 103–124.

LARKIN, J., and H. SIMON. 1987. Why a diagram is sometimes worth 10,000 words. Cognitive Science, **11**: 65–99.

LARIZZA, C., A. MOGLIA, and M. STEPHANELLI. 1992. M-HTP: A system for monitoring heart-transplant patients. Artificial Intelligence in Medicine, **4**: 111–126.

MACKINLAY J., G. G. ROBERTSON, and S. K. CARD. 1994. The perspective wall: Detail and context smoothly integrated. *In* Proceedings of CHI '94, pp. 173–179.

MCDERMOTT, J. 1988. Preliminary steps toward a taxonomy of problem-solving methods. *In* Automating Knowledge-Acquisition for Expert Systems. *Edited by* S. Marcus. Boston: Kluwer, pp. 225-256.

MIKSCH, S., Y. SHAHAR, and P. D. JOHNSON. 1997. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. *In* Proceedings of the Seventh Workshop on Knowledge Engineering Methods and Languages KEML-97, Open University, Milton Keynes, UK.

MOLINA, M., and Y. SHAHAR. 1996. Problem-solving method reuse and assembly: From clinical monitoring to traffic control. *In* Proceedings of the Tenth Banff Knowledge Acquisition for Knowledge-based systems Workshop, Banff, Alberta, Canada, Vol. 1, pp. 7-1—7-20.

MUSEN, M.A. 1989. Automated Generation of Model-based Knowledge-Acquisition Tools. Morgan Kaufmann, San Mateo, CA.

MUSEN, M.A. 1992. Dimensions of knowledge sharing and reuse. Computers and Biomedical Research **25**, 435–467.

MUSEN, M.A., J. H. GENNARI, H. ERIKSSON, S. W. TU, and A. R. PUERTA. 1995. PROTÉGÉ-II: Computer support for development of intelligent systems from libraries of components. *In* MEDINFO '95: Proceedings of the Eighth World Congress on Medical Informatics, Vancouver, British Columbia, pp. 766–770.

MUSEN, M. A., S. W. TU, A. K. DAS, and Y. SHAHAR. 1996. EON: A component-based approach to automation of protocol-directed therapy. Journal of the American Medical Association, **3**(6): 367–388.

MUSEN, M. A. 1998. Domain ontologies in software engineering: Use of Protégé with the EON architecture. Methods of Information in Medicine, **37**: 540-550.

NEWELL, A. 1982. The knowledge level. Artificial Intelligence, **18**: 87–127.

NGUYEN, J., Y. SHAHAR, S. W. TU., A. K. DAS, and M. A. MUSEN. In press. Integration of temporal reasoning and temporal-data maintenance into a reusable database mediator to answer abstract, time-oriented queries: The Tzolkin system. Journal of Intelligent Information Systems.

PLAISANT, C., B. MILASH, A. ROSE, S. WIDOFF, and B. SHNEIDERMAN. 1996. Lifelines: Visualizing Personal Histories. Proceedings of CHI '96, Vancouver BC, pp. 221–227.

POWSNER, S. M. and E. R. TUFTE. 1994. Graphical Summary of Patient Status. Lancet, **344**: 386–389.

PUERTA, A. R., J. W. EGAR, S. W. TU, and M. A. MUSEN. 1992. A multiple-method knowledge-acquisition shell for the automatic generation of knowledge-acquisition tools. Knowledge Acquisition, **4**: 171–196.

PUERTA, A. 1997. A model-based interface development environment. IEEE Software, **14**(4): 40–47.

RAO, R., S. CARD, H. JELLINEK, J. MACKINLAY, and G. ROBERTSON. 1992. The Information Grid: A Framework for Information Retrieval and Retrieval-Centered Applications. *In* Proceedings of the ACM Symposium on User Interface Software and Technology '92, Monterey, CA, pp. 23–32.

ROHRER, R., and E. SWING. 1997. Web-based information visualization. IEEE Computer Graphics and Applications, **7/8**: 52–59.

ROTH, S. F., J. KOLOJEJCHICK, J. MATTIS, and J. GOLDSTEIN. 1994. Interactive graphic design using automatic presentation knowledge. *In* Proceedings of CHI '94, pp. 112–117.

RUSS, T.A. 1989. Using hindsight in medical decision making. *In* Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care, Washington, D.C. *Edited by* L. C. Kingsland, IEEE Computer Society Press, Los Alamitos, CA, pp. 38–44.

SHAHAR, Y., S. W. TU, and M. A. MUSEN. 1992. Knowledge acquisition for temporal-abstraction mechanisms. Knowledge Acquisition, **4**: 217–236.

SHAHAR, Y., and M. A. MUSEN. 1993. RÉSUMÉ: A temporal-abstraction system for patient monitoring. Computers and Biomedical Research, **26:**, 255–273. *Reprinted in* Yearbook of Medical Informatics 1994. *Edited by* J. H. van Bemmel, and A. T. McRay, F.K. Schattauer and The International Medical Informatics Association, Stuttgart, Germany, pp. 443–461.

SHAHAR, Y., and M. A. MUSEN. 1996. Knowledge-based temporal abstraction in clinical domains. Artificial Intelligence in Medicine **8** 3, 267–298.

SHAHAR, Y. 1997. A framework for knowledge-based temporal abstraction. Artificial Intelligence, **90**(1–2): 79–133.

SHAHAR, Y. 1998. Dynamic temporal interpretation contexts for temporal abstraction. Annals of Mathematics and Artificial Intelligence, **22**(1–2): 159–192.

SHAHAR, Y., and M. MOLINA. 1998. Knowledge-based spatiotemporal abstraction. Pattern Analysis and Applications, **1**(2): 91-104.

SHAHAR, Y., S. MIKSCH, and P. D. JOHNSON. 1998. A task-specific ontology for the application and critiquing of time-oriented clinical guidelines. Artificial Intelligence in Medicine, **14**: 29-51.

SHAHAR, Y. 1999. Knowledge-based temporal interpolation. Journal of Experimental and Theoretical Artificial Intelligence, **11**: 123-144.

SHAHAR Y., H. CHEN, D. P. STITES, L. M. BASSO, H. KAIZER, D. M. WILSON, and M. A. MUSEN. In press. Semiautomated acquisition of clinical temporal-abstraction knowledge. The Journal of the American Medical Informatics Association.

SHNEIDERMAN, B. 1994. Dynamic Queries for Visual Information Seeking. IEEE Software, **11**(6): 70–77.

SHOHAM, Y. 1987. Temporal logics in AI: Semantical and ontological considerations. Artificial Intelligence, **33**: 89–104.

SNODGRASS, R., and I. AHN. 1986. Temporal databases. IEEE Computer, **19**: 35-42.

STEIN, A., M. A. MUSEN, and Y. SHAHAR. 1996. Knowledge acquisition for temporal abstraction. Proceedings of the 1996 AMIA Annual Fall Symposium formerly the Symposium on Computer Applications in Medical Care, Washington, DC, pp. 204–208.

TUFTE, E. R. 1982. The Visual Display of Quantitative Information. Graphics Press, CT.

TUFTE, E. R. 1990. Envisioning Information . Graphics Press, CT.

TUFTE, E. R. 1997. Visual Explanations. Graphics Press, CT.

TU, S. W., M. G. KAHN, M.A. MUSEN, J.C. FERGUSON, E.H. SHORTLIFFE, and L.M. FAGAN, Episodic skeletal-plan refinement based on temporal data, Communications of the ACM **32**12 1989 1439–1455.

TU, S. W., Y. SHAHAR, J. DAWES, J. WINKLES, A. R. PUERTA, and M. A. MUSEN. 1992. A problem-solving model for episodic skeletal-plan refinement. Knowledge Acquisition, **4**: 197–216.

TU, S. W., H. ERIKSSON, J. H. GENNARI, Y. SHAHAR, and M. A. MUSEN. 1995. Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: Application of PROTÉGÉ-II to protocol-based decision support. Artificial Intelligence in Medicine, **7**(3), 257–289.

WIEDERHOLD G., R. L. BLUM and M. G. WALKER. 1986. An integration of knowledge and data representation. *In* On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies. *Edited by* M. L. Brodie, J. Mylopoilos, and J. W. Schmidt, Springer-Verlag, pp. 431–444.

WIEDERHOLD G. 1992. Mediators in the architecture of future information systems. IEEE Computer, **3**: 38–49.

WEILINGA, B., A. T. SCHREIBER, and J. BREUKER. 1992. KADS: a modeling approach to knowledge engineering. Knowledge Acquisition, **4**: 5–53.

WRIGHT, W. 1997. Business visualization applications. IEEE Computer Graphics and Applications. **7/8**: 66–70.

ZHOU, M., and S. FEINER. 1996. Data characterization for automatically visualizing heterogeneous information. *In* Proceedings of Information Visualization '96, pp. 13–20.