Navigation and Visualization of Abstractions of Time-Oriented Clinical Data

Cleve Cheng, Yuval Shahar, Angel Puerta
Section on Medical Informatics
MSOB x215, Stanford University
Stanford, CA 94305-5479 USA
+1 650 723 6979
cleve@cs.stanford.edu,
{shahar,puerta}@smi.stanford.edu

Daniel Stites
Department of Laboratory Medicine
University of California, San Francisco
San Francisco, CA 94143-0134 USA
+1 415 476 1186
stites@labmed2.ucsf.edu

ABSTRACT

We describe the methodology and architecture of a knowledge-based, interactive visualization system that enables physicians and medical support personnel to draw conclusions from heterogeneous time-oriented clinical data. Our system employs domain-specific ontologies to produce temporal and statistical abstractions of data, and also as the basis for semantically-based browsing and visualization. This builds on previous work in data mining, temporal reasoning, and information visualization, but offers fundamental advantages over any isolated approach, by leveraging each off the others. We performed an evaluation of a prototype, leading us to conclude that users can indeed use the system to perform such semantically-based browsing in a reasonable amount of time.

Keywords

Information analysis, information visualization, health care, decision support, model-based interface development, dynamic query, temporal abstraction, knowledge bases

INTRODUCTION

While medical informatics has come a long way in harnessing computational power to display spatially oriented clinical data (e.g., volume rendering CT scans), this type of clinical data represents only a small fraction of the amount of information available on a patient. For non-spatially-oriented data like blood test parameters, attending physicians and medical support personnel regularly wrestle with paper-based patient charts.

A more generally applicable perspective from which to view clinical data is as a heterogeneous set of time series. That is, many domains within health care require decisions based on recognition of temporal patterns within patients' records. As it is a long and tedious process for doctors to filter the data by hand in order to isolate particular problems, it is an ideal task for a computer to perform this abstraction of data into more useful forms. This abstraction of data is highly idiosyncratic, and relies on the vast store of knowledge required of doctors. Thus, a domain-specific knowledge base must be developed which captures the forms and parameters of this abstraction process.

Once this knowledge base is in place and abstraction of patient data has been accomplished, the next step is to visualize and navigate through this information. Because doctors are used to performing these abstractions themselves, they will tend to distrust the "conclusions" drawn by the computer, and thus also seek an "audit trail"—detailed explanations for the abstractions they see on the screen. They want to see both the raw data on which the abstractions are based and the mechanism by which they are derived. In order to discern temporal patterns and glitches in the audit trail, they want to make comparisons between various abstraction levels. This is more than just "drilling down" in the database sense; this is the type of information that is usually found in a knowledge base. Thus, both the temporal abstraction itself and the navigation require a domain-specific knowledge base.

Our project's purpose is to provide this kind of richer navigation. The target groups for our interface design are physicians and medical support personnel, each encompassing a wide range of computer and computer programming experience. However, we believe the architecture is general enough to encompass any domain in which temporal abstraction is of prime importance.

Outline

We begin with a description of previous work in related areas. After that, we proceed to a description of our solution and the development cycle for our interface, all in the context of an example application in the domain of protocol-based patient care. Following this is a description of our evaluation of a prototype implementation, and results of our testing. Finally, we discuss possible extensions of our system and areas of further research.

RELATED WORK

Ahlberg and Shneiderman's starfield displays [1] and Lamping, Rao, and Pirolli's hyperbolic trees [6] represent attempts to navigate through graph structures or databases using spatial metaphors. These techniques are useful for emphasizing and browsing relations between data objects, but have limited applicability when a primary object of visualization is the information contained *within* the objects, as is the case with medical data.

Other work in focusing lenses and movable filters [3] addresses the visualization of non-spatial or textual data. This is useful for approaches such as ours, but falls short of the domain-specific interpretations that provide our system with its increased explanatory power.

Zhou and Feiner [13] give a taxonomy of data characteristics for visualization, some of which is duplicated in our model. However, by concentrating on temporal relations and adding domain knowledge, our data characterization scheme is more detailed, and can therefore display and filter on more specific relations of interest to the user.

Golovchinsky et. al. [5] describe a system which automatically generates graphics such as timelines based on domain knowledge. The system uses knowledge much the same way as our visualization module does, but stops short of employing this knowledge for navigation as well.

Our visualization technique draws on previous work by Cousins and Kahn, and Plaisant and her colleagues. Cousins and Kahn [2] described a graphical semantics for certain operations on temporal intervals that we employ in our visualization process, but did not address the problems of navigating temporal intervals.

In their Lifelines project, Plaisant, et. al. [7], present a visualization technique that bears a striking resemblance to our own. However, they do not implement the semantic navigation capabilities that lie at the core of our system nor does is their system capable of automatically forming temporal abstractions of patient data.

SOLUTION & APPLICATION

Our system is called KNAVE: Knowledge-based Navigation of Abstractions for Visualization and Explanation. The knowledge base, abstraction generator, navigation/explanation engine, and visualization unit are each separate modules (see Figure 1) which we will describe in turn.

In order to provide some context to our description of this system, we first describe an example sub-domain of health care on which we based our evaluation: protocol-based patient care.

Protocol-based patient care

In medical clinics, protocols are used regularly in the treatment of seriously ill patients. Often they are developed because the treatment is experimental or so dangerous to the patient that they are only slightly less harmful than the diseases they are designed to treat. They are agreed upon by a body of experts in the particular field of medicine to which it applies. They include schedules for the administration of drugs, schedules for the performing of tests, indicators of danger to the patient, and procedures in case something goes wrong. The schedules and indicators are what we attempt to model in our ontology, providing a tool to support a doctor's decision to take someone off the protocol or initiate a change in protocol.

Résumé and knowledge bases

Our system abstracts data using a temporal inference engine called Résumé [9] and its associated architecture. Résumé forms its abstractions according to the knowledge structures of a given domain. Such a structure is called an *ontology* and is defined by a domain expert working with a *knowledge acquisition tool*, a tool for entering an ontology into a program.

A Résumé ontology is specially-constructed for the purpose of temporal abstraction. Elements in the domain are grouped into one of three categories: contexts, actions, and parameters. Parameters are generally measured data and abstractions of that data. They can take on discrete or continuous values. Like parameters, actions can have different values, and can be abstracted into other actions. For instance, a shot of insulin has a particular dosage which can serve as its value, and may be part of a treatment which is treated as another event. Finally, contexts are periods of situations in which abstract parameters and actions have a certain interpretation. An example of this is the context of the effects of chemotherapy, during which lab data which may normally be considered alarming are considered part of the normal course of treatment. Contexts can have sub-contexts, such as the period of effect of a certain drug during a chemotherapy protocol.

The Domain Ontology Server (see Figure 1) provides the ontology to KNAVE. One of the components of the Domain Ontology Server is the Temporal Abstraction and Visualization (TAV) Server, which provides knowledge about how to make abstractions in this domain, and various parameters to optimize visualization of these abstractions. An example of temporal abstraction knowledge is the classification function from a parameter value to the value of a higher-level abstraction.

Résumé is encompassed within a *temporal mediator* [12] called Tzolkin. The term temporal mediator is a short way of saying that Tzolkin mediates between a database and a program that makes temporal queries (KNAVE in this case). The answer to the query might involve computing abstractions not held in the database, in which case it uses a temporal abstraction module (Résumé in this case) to produce these abstractions. Tzolkin also contains a module, called Chronus, for performing temporal pattern matching directly on the abstractions contained in the Tzolkin database. Thus, Tzolkin also has a module which processes

SQL or Résumé-style queries, and refers it to Chronus, Résumé, or, if the query is immediately answerable, to the database query engine itself. In terms of our conceptual framework, Tzolkin is a Temporal and Statistical Abstraction (TSA) Server, meaning that it provides the particular values and time intervals of the various abstraction types to KNAVE.

KNAVE architecture

With the help of the TSA and Domain Ontology Servers, KNAVE implements the visualization and navigation/explanation functions of the system. The computational module handles the navigation and explanation, while the graphical interface determines the visualization of the data (see Figure 1). The graphical interface was developed as a separate entity, in order to maintain adaptability to a wide range of domains through modular design.

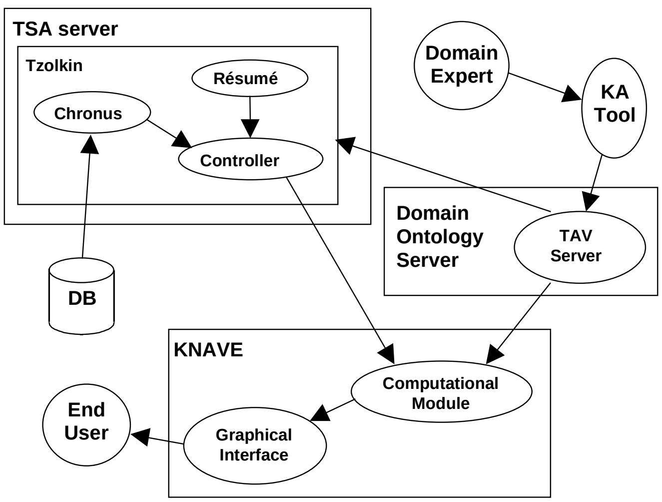

Figure 1. The architecture of KNAVE. TAV stands for Temporal Abstraction and Visualization. TSA stands for Temporal and Statistical Abstractions. DB refers to the patient database of raw facts. KA stands for Knowledge Acquisition.

Computational module

The computational module integrates the knowledge obtained from queries to the TSA and Domain Ontology Servers. It maps widgets to queries, and query results to widgets. Its navigation model is described later in the section entitled "Integration of User Task and Domain Models."

Using MOBI-D

In developing this interface, we followed a model-based approach, using the MOBI-D (MOdel-Based Interface Designer) [8] interface development environment. The MOBI-D development cycle has three stages. The first stage is the elicitation of user task and domain models. The second stage is the integration of these two models to produce a framework for the interface. The third stage is the construction of a presentation and dialog. We describe each stage in the process of building our current system.

Elicitation of user task and domain models: user task

The user task model was developed using a module of the MOBI-D environment called U-TEL [11]. A potential user of the system enters a plain text description of the task they wish to accomplish with the interface, and then proceeds to create an outline of its basic components. This outline is then used as a specification for the user task model.

Some key aspects of the user task model we developed with this tool are: navigation along multiple dimensions derived from an ontology of temporal abstraction, frequent communication of the underlying temporal inference processes to the user, and the need for statistical as well as temporal navigation and abstraction.

The navigation dimensions are divided into two categories: *syntactic navigation* and *semantic navigation*. Syntactic navigation covers all forms of navigation that do not require domain knowledge. There are several forms of syntactic navigation with which we are familiar: zooming and scrolling in the time dimension, zooming individual charts along the parameter value dimension, scrolling through charts already displayed, and collapsing and expanding of graph hierarchies and ontology components.

Temporal semantic navigation is the key innovation of our system. There are three dimensions to navigate along for both parameters and actions, and they are independent of domain.

The first involves moving up and down relations in the domain's ontology, such as abstraction hierarchies. We refer to this as *semantic drill-down*. Referring to Figure 3, a user might see Bone Marrow Toxicity values and wonder what information they were derived from. She would navigate down the abstraction hierarchy, seeing Granulocyte State values, and then values for Granulocytes. She finds she has reached raw values, and cannot drill any further.

The second involves changing the time unit (temporal granularity). In many systems, this just re-labels the x-axis, but KNAVE uses domain knowledge to determine whether it is more appropriate to show a statistical abstraction instead of the values themselves. For instance, a user switches the time unit from hours to years. Suddenly, the blood sugar measurements (taken several times a day) look more like noise than the discrete values he was looking at. It becomes more useful to show a distribution of values for each year, indicating mean, variance, and other statistical parameters. The knowledge of what temporal granularity change to make this adjustment and which surrogate abstraction to use depends on the domain, even the particular parameter in the domain (e.g., it makes sense to look at individual height values at the granularity of years).

The third dimension involves changing contexts. A user may want to view a given abstract parameter, say White Blood Cell Toxicity, within different contexts to get an idea of what the patient's status is under various simultaneous treatments. This gives the doctor an idea of the interaction between treatments. These dimensions define non-Euclidean spaces, and as such, are difficult to navigate using spatial metaphors.

Elicitation of user task and domain models: domain

In the production of the domain model, we were careful to keep separate three domains. The first is the visualization, navigation, and explanation of abstractions. The second is the formation of abstractions of time-oriented data. The third is the formation of abstractions of *medical* time series data.

The first model we developed from doctors using U-TEL. The second model, the domain model for temporal abstraction, is the core of the Résumé system, and thus already existed. The third model was elicited in a separate stage, through the use of a knowledge-acquisition tool tailored for use with Résumé. In practice, both the general knowledge common to all medical applications (drugs, tests, etc.) and the specific ontology for protocol-based care was acquired at the same time, but they are theoretically distinct, and could just as easily have been acquired as separate ontologies.

Integration of user task and domain models

The integration of the two models pivots around the concept of *semantic zoom operators* [2]. The Résumé temporal-abstraction system determines possible dimensions along which to explore abstraction levels. The task model determines typical actions and types of information the user needs. We interpret these actions as sequences of simultaneous steps along various dimensions to explore. The process of developing an integrated model is then reduced to the construction of all necessary sequences, and each sequence is defined as a semantic zoom operator and mapped to interface components.

We integrate the domain model concept of sending queries to a temporal inference system with the task model concept of moving along various dimensions of abstraction by simply treating them as one and the same thing. Shneiderman describes this *dynamic query* method in [10]. It involves sending a series of rapid queries as users move along a navigation axis. Ahlberg and Shneiderman [1], Fishkin and Stone [3], and Goldstein and Roth [4] have all used this method for interactive navigation tasks. Unlike previous applications of the dynamic query method, KNAVE does not need to navigate continuous dimensions, but need only jump between discrete levels of abstraction. The assumption is that users will not need to rapidly flip through many levels of abstraction at animation speeds, as different levels of abstraction seldom have the level of visual coherency necessary to make this a profitable action.

Constructing An Interface

The interface (see Figure 2) is composed mainly of two entities: tree browsers and time chart objects. These correspond to the two elements of domain knowledge and temporal abstraction contained within Résumé's domain

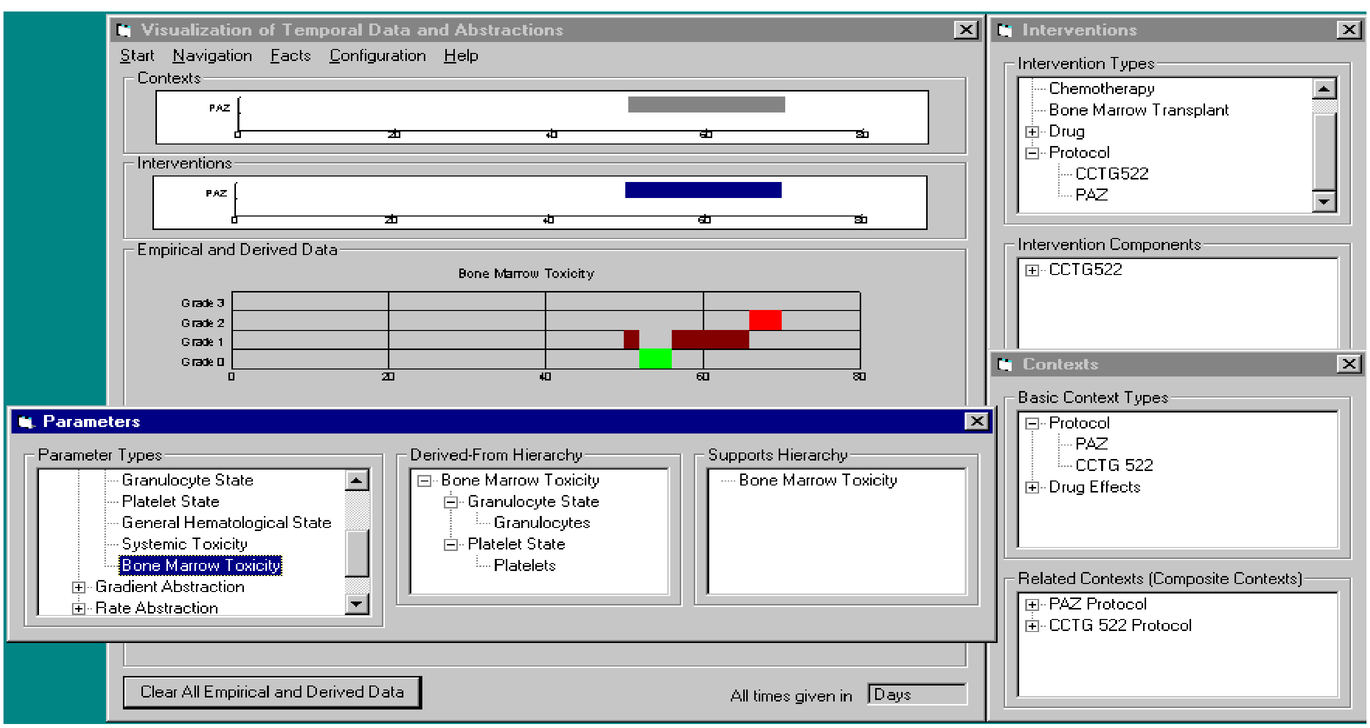

Figure 2. The KNAVE interface, showing the Parameter, Intervention, and Context navigation windows and the main window, showing the parameter Bone Marrow Toxicity in the context of the PAZ treatment protocol.

model.

Six to eight small tree browsers (depending on user configuration settings) present different views on the context, event, and parameter ontologies, allowing the user a wide variety of navigation possibilities by simply clicking on one of the words in any tree browser. In each navigation window, there are one or two browsers for showing relations between parameters at various abstraction levels. This enables the user to drill-down using the semantics of the domain, which is a major aspect of the user task model.

Time charts are the atomic graphical entities that present all abstractions and time-oriented data in KNAVE. They are composed of four types: time interval charts for displaying discrete abstractions, scatter plots for displaying raw data, hybrid charts for displaying continuous abstractions, and histograms and other specialized charts for displaying statistical information.

A chart of any of these types can perform a number of operations. These include scrolling both along the time axis and the value axis, magnifying in either direction, reordering itself among the currently displayed charts, and jumping to the next level of abstraction along a variety of semantic axes. In addition, charts of the first three types (what we call temporal charts) can be overlaid or combined with any other temporal chart.

The charts are composed in a panel in the main window (Figure 2). There are three panels in this window: the Context panel, the Action (or Interventions) panel, and the Parameter (or Empirical and Derived Data) panel. These map to the fundamental division between context, actions, and parameters in the domain model. The panel currently being explored is enlarged to take up most of the window, while the others are shrunk to show a minimal amount of information. This is an implementation of change of focus between actions and parameters, as described in the user task model. Within each panel, graphs are stacked vertically. They remain temporally aligned to allow users to make comparisons between different parameter values. Again, this is one of the key requirements of the user task model.

EVALUATION

After completing a prototype of the system, we evaluated it in order to assess the viability of our knowledge-based approach, and elicit suggestions for future versions.

We asked seven potential users to evaluate our system. The group included both experienced physicians with decades of clinical experience, a student who had just finished his residency, medical students, and a medical informatician with medical knowledge comparable to that of typical medical support staff. The group also represented both experienced computer programmers as well as casual users of only a handful of applications. The group included both male and female participants.

Each potential user was given an introduction to the main features of the system (ten minutes, plus time for questions). They were then asked to perform three short tasks, and provide any comments they might have about problems with the system or modifications of the design. A single patient file was constructed from actual cases of administrations of an AIDS treatment protocol and bone marrow disease protocol, and users were asked to extract information about this patient.

The tasks were designed to cover a wide range of activities possible with the KNAVE system, including all of the most common ones. Thus, one task was to find "if an occurrence of at least one week of at least Grade 2 Bone Marrow Toxicity (in the PAZ graft-versus-host disease protocol context) exists in the abstractions of a particular patient's data, and if so, on which data is that abstraction based." Bone Marrow Toxicity is a higher-level abstraction that characterizes the effect of chemotherapy on the bone marrow. PAZ stands for prednisone/azathioprine, two drugs used in the protocol, and is a treatment for chronic graft-versus-host disease, a complication of a bone marrow transplant. This query is typical of the kind given in the elicited user task model. It forces the users to use the explanatory capabilities of the KNAVE system. One stage in the completion of this task is shown in Figure 3.

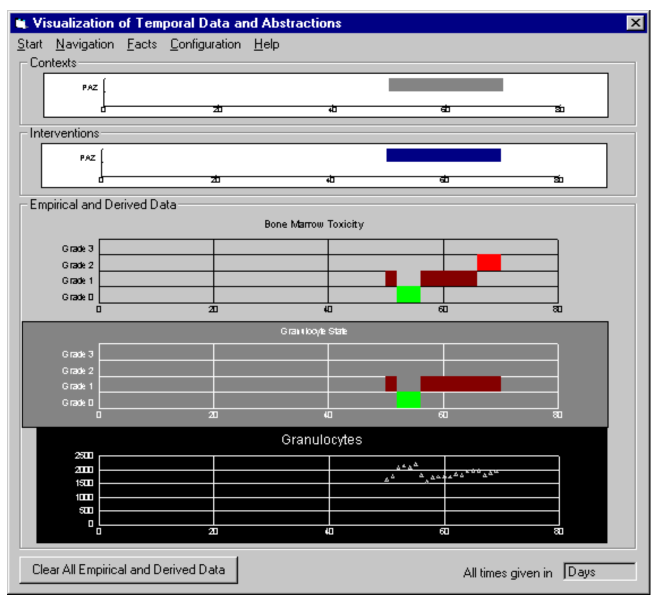

Figure 3. In the PAZ context, Bone Marrow Toxicity values are derived from Granulocyte State values, which are derived from values of Granulocytes.

A second task emphasized the context-changing mechanism of the system. The question put to the users was, "can you find any intervals of pancytopenia or polycythemia as expressed by the parameter 'General Hematological State' in the context of the administration of the CCTG AIDStherapy protocol?" The idea here is to see if users could grasp that the context changed, as well as how to change it. Pancytopenia and polycythemia are abnormal values for this parameter. General Hematological State is a summary of all the hematological (blood-related) data measured for the purposes of the treatment. This parameter and supporting information is shown in Figure 4.

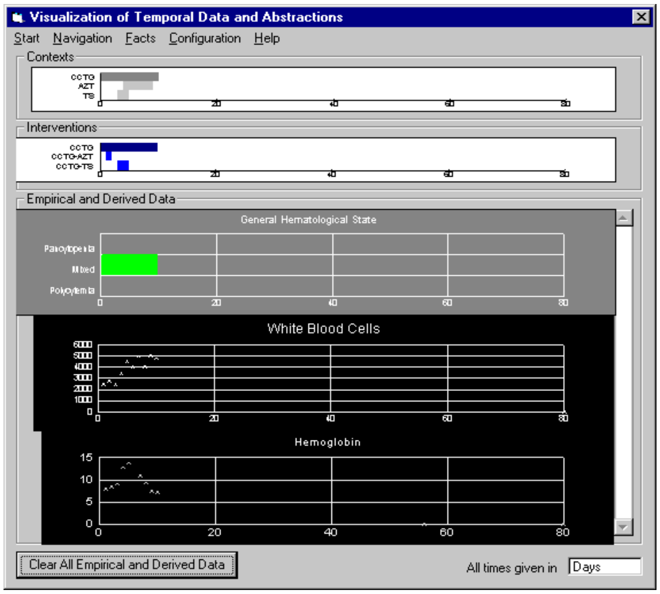

Figure 4. In the CCTG context, General Hematological State is derived from two parameters: White Blood Cells and Hemoglobin.

The final task was to "find the distribution of values of the parameter 'Hemoglobin State' in the CCTG context over the last 80 days." The purpose of this last test was to highlight the statistical abstraction (contrast with temporal abstraction), visualization, and navigation aspects of KNAVE. Hemoglobin State is a categorization of hemoglobin measurements (hemoglobin is the oxygen-binding protein contained in red blood cells).

Efficiency observations

Not only were all users able to find the requested information without guidance, but they were also able to do it quickly. Completion times for the sum of the three tasks ranged from 20 seconds to 3 minutes, with six of the users completing the tasks in under 90 seconds. For six of the users, the first task took the longest, roughly half the time, and the second almost as long. The last user took longer to perform the second task than the first. All seven took only a few seconds to complete the third task.

It would be unfeasible to compare these times to completion times for human-knowledge-based abstraction and decision-making. The reason is that few doctors are familiar enough with these experimental protocols to interpret the raw data in any reasonable amount of time. For our example domain, there are six raw data types and fourteen abstraction types, with one-to-three-dimensional derivation tables for each. Clearly, the time it takes to learn a new protocol and mechanics of forming various abstractions is far longer than the 3 minutes the slowest user of our system took to perform the tasks. Similarly, even given a tool for computing and visualizing temporal abstractions, but without the semantic navigation capabilities of KNAVE, non-experts lack the knowledge of how abstract parameters are related. Navigating in this way requires users to pick from a list of unorganized parameters, making semantic drill-down impossible for the non-expert. For the first task, the one emphasizing the drill-down mechanism, the slowest of our seven users navigated in a similar way, each time choosing new parameters from lists of raw data and abstract parameters, rather than following abstraction links. Even so, this user still benefited from *some* structuring of the data provided by the domain model, and thus was better off than someone without a navigation system at all.

Even if we were to find enough experts to carry out such a comparison, they would then fail to represent our entire target group. One of the goals of KNAVE is to provide decision support for doctors who lack the extensive experience with a given protocol to be able to interpret patient data on the run, yet have sufficient general knowledge to understand the medical semantics behind the explanations provided by the system.

Summary of comments

Users were enthusiastic about the features of the system, with some emphasizing its explanatory capabilities. Two commented that the statistical abstractions provided much needed data.

Other comments seemed to emphasize the advantages of multiple navigation modes. We found that each user followed a different path through the dialog. Anticipating the possibility of widely varying needs for different user groups, we chose flexibility over consistency in many of our design decisions. The rationale is that various user groups can be identified and profiles developed which capture the preferred navigation mode for those groups. In the prototype used for these evaluations, we did not specialize for particular groups, and therefore the system still maintained maximum flexibility. We believe the short time-to-completion for the tasks given is related to the users' ability to choose whichever mode of navigation they learned the quickest.

Trends

The particular paths different users took through the dialog did not seem to have any correlation to the particular demographic group divisions we defined at the start of this Evaluation section. While it is always questionable to talk about correlation with a sample set of seven, it was nevertheless obvious that there were five distinct methods of navigating the interface among our seven users. One used the Query Dialog exclusively, one used the query dialog only once and the tree-browser interface the rest of the time (except for the third task which required the use of the popup menu), one used only the pop-up menus and the Query Dialog. Two used the context tree-browser to change contexts, but otherwise used the pop-up menus to navigate. Finally, two used a mixture of the various navigation methods to perform the first two tasks.

Another interesting trend which emerged was the ability of the five younger users (all in their twenties) to complete each task significantly quicker than the two older users (experienced physicians in their fifties). This points to at least partial success in our goal of providing decision support for less experienced practitioners. After the evaluation, one of the younger users speculated that the older doctors have had longer to get used to inspecting scatter plots of primitive data directly, and thus are less susceptible to having their conclusions handed to them. However, this is only speculation, as the evaluation was not designed to draw such conclusions.

EXTENSIONS

Application to other domains

We have already developed ontologies for other domains, both medical domains like children's growth, diabetes, and non-medical ones like traffic control. We have also successfully formed temporal abstractions of data using these ontologies. These three domains will stretch the capabilities of our current implementation of KNAVE.

Diabetes and smart databases

Diabetes is different from protocol-based care in that, instead of a large number of raw data parameters and a couple levels of abstraction of those parameters, it has very few parameters (typically just blood sugar levels) and a few actions (like insulin administration and mealtimes), but a complex set of abstractions, involving periodic patterns (such as low blood sugar levels on weekends only). Furthermore, since diabetes patients typically take blood glucose measurements four times daily, databases that cover whole life histories are large. Thus, we are looking into problems of scalability with our current implementation. The solution to these problems is to shift part of the load of performing temporal inferences to the database itself—a smart database approach [12].

Children's growth and domain-specific graphics

Children's growth monitoring brings up limitations in our graphical representation of data. While we believe our four graph types capable of displaying any type of time-oriented data for which an ontology exists, another issue to consider is whether they present information in the most intuitive way.

Even physicians who do not specialize in children's growth are familiar with the type of graphic known as a growth chart (an example is shown in Figure 5). The y-axis corresponds to a common measure of a child's growth (in this case height), while the x-axis shows the child's age. Several arcs of solid or dashed lines show growth curves for various percentile ranges (percentile given at the bottom left of each curve). For instance, 97% of all Chinese girls of the sample set used to make this chart had heights on or below the top curve throughout ages 6 through 18. The location and spacing between these growth curves are adjusted by restricting the sample set according to factors such as gender, race, and the previously observed growth curve of the particular child. The physician can tell at a glance how the child is doing by simply locating the child's height and age on the grid.

As you can see, such a chart is fundamentally different from the scatter plots and timelines we use in our system. While the x-axis still represents time, the chart itself does not represent the patient in question, but rather a complex statistical abstraction that determines typical ranges for a population of which the patient is a part. As such, our patient-centered data model cannot be made to display such a chart without further specialization of the interface to the domain of children's growth monitoring.

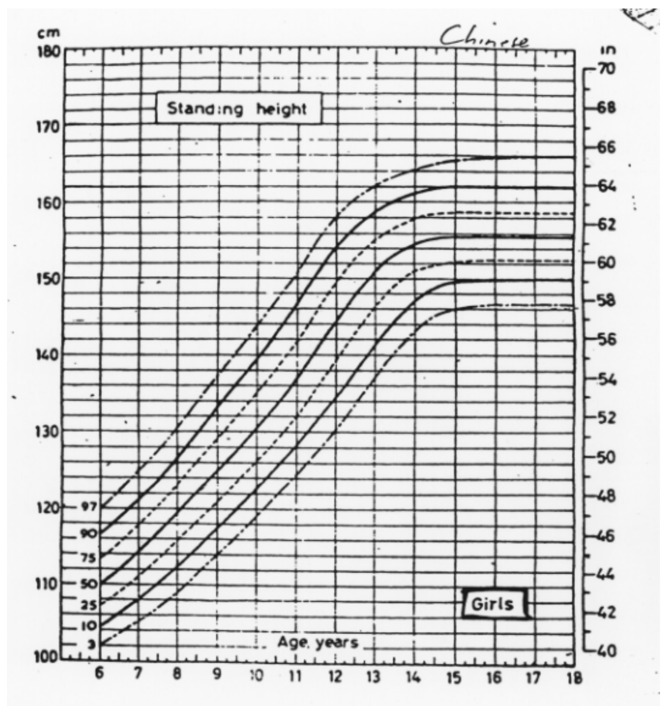

Figure 5. Growth chart for Chinese girls.

Traffic control and multi-dimensional navigation

Traffic control requires the ability to make abstractions both in time and across distances. An example is the attempt to spot a traffic jam as it begins to develop, but before all traffic flow has stopped. One must not only consider car velocities over given time intervals, but also over different stretches of road. Strictly speaking, KNAVE is neither restricted to the domain of health care nor to the methodology of temporal reasoning. When applying the Résumé system to the domain of traffic control, we refer to it as a *linear* abstraction method. This is because it doesn't matter what physical dimension Résumé is forming abstractions along—it could be the distance along a highway just as easily as time.

Of course, this presents challenges for navigation and visualization. While it is perfectly possible to display such information in the current system, it is not clear that this is the most intuitive or powerful way. Intervals of time could be placed alongside intervals of distance, but simultaneous two-physical-dimensional navigation is not currently within the system's abilities. Here it is important to distinguish between navigating physical dimensions and navigating *semantic dimensions*. The latter refers simply to the three dimensions of non-Euclidean navigation described in the earlier User Task Model section. To navigate two physical dimensions simultaneously, KNAVE would need to navigate *six* dimensions simultaneously (assuming actions and parameters are not simultaneously navigated). We need a method of visualizing n-physical-dimensional data (where one or more of these dimensions are more important than others) while semantically navigating (n-1)-physical-dimensional information spaces. We need to navigate one less dimension than we visualize, because we need not semantically navigate the parameter value dimension. For example, in the traffic control domain, we would ideally visualize three dimensions (time, location along roads, and a third parameter such as the velocity of automobiles), but would only need to simultaneously navigate in the two dimensions of time and location.

Need detailed domain knowledge

The system relies heavily on detailed knowledge provided by a domain expert. Seen another way, this is where the system derives its navigational fluency and explanatory ability. The ontology only needs to be entered once, and once entered, can be tweaked to accommodate individual doctors' preferences. We see no way to achieve this level of fluency without the help of a domain expert.

CONCLUSIONS

We have developed a system that performs knowledge-based abstraction of time-oriented information, visualizes these abstractions with their raw data, and enables semantically-based navigation of that information. This architecture provides doctors and medical support personnel the ability to draw conclusions from patient data in a framework defined by other doctors. Our evaluation has shown that it properly enables such a navigation method, and presents information in a more quickly comprehensible form than an existing method of interpreting clinical data.

ACKNOWLEDGMENTS

This work has been supported by grants LM05708 and LM06245 from the National Library of Medicine and IRI-9528444 from the National Science Foundation. We thank Larry Basso and Darrell Wilson for insight on the needs of clinicians, and Chaohuang Zeng and Hai Chen for their work on adapting the Résumé system to the needs of our visualization/navigation task.

REFERENCES

1. Ahlberg, C. and Shneiderman, B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. *Proceedings of CHI '94* (Boston MA, Apr 1994), ACM Press, 313-317.
2. Cousins, S. and Kahn, M. The Visual Display of Temporal Information. *Artificial Intelligence in Medicine 3* (1991), 341-357.
3. Fishkin., K. and Stone, M. Enhanced Dynamic Queries via Movable Filters. *Proceedings of CHI '95* (Denver CO, May 1995), ACM Press, 415-420.
4. Goldstein, J. and Roth, S. Using Aggregation and Dynamic Queries for Exploring Large Data Sets. *Proceedings of CHI '94* (Boston MA, April 1994), ACM Press, 23-29.
5. Golovchinsky, G., Kamps, T., and Reichenberger, K. Subverting Structure: Data-driven Diagram Generation. *Visualization '95* (Atlanta GA, Oct-Nov 1995), IEEE Computer Society Press, 217-223.
6. Lamping, J., Rao, R., and Pirolli, P. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Proceedings of CHI '95* (Denver CO, May 1995), ACM Press, 401-408.
7. Plaisant, C., Milash, B., Rose., A, Widoff, and S., Shneiderman, B. Lifelines: Visualizing Personal Histories. *Proceedings of CHI '96* (Vancouver BC, April 1996), ACM Press, 221-227.
8. Puerta, A. A Model-Based Interface Development Environment. *IEEE Software* 14, 4 (Jul/Aug 1997), 40-47.
9. Shahar, Y. A Framework for Knowledge-Based Temporal Abstraction. *Artificial Intelligence* 90 (1-2), 79-133.
10. Shneiderman, B. Dynamic Queries for Visual Information Seeking. *IEEE Software* 11, 6 (Nov 1994), 70-77.
11. Tam, R., Maulsby, D., and Puerta, A. A Tool for Eliciting User Task Models from Domain Experts. *Proceedings of IUI '98* (San Francisco CA, Jan 1998), (in press).
12. Wiederhold, G. Mediators in the Architecture of Future Information Systems. *IEEE Computer* 3 (1992), 38-49.
13. Zhou, M. and Feiner, S. Data Characterization For Automatically Visualizing Heterogeneous Information. *Information Visualization '96*, IEEE Computer Society Press, 13-20.