A Temporal-Abstraction Mediator for Protocol-Based Decision Support

Amar K. Das, Yuval Shahar, Samson W. Tu, and Mark A. Musen
Section on Medical Informatics, Stanford University School of Medicine
Stanford, California 94305-5479 (das@camis.stanford.edu)

Abstract

The inability of many clinical decision-support systems to integrate with legacy databases limits the wide-scale deployment of such systems. To overcome this obstacle, we have designed a general data-interpretation module that can be embedded in a comprehensive architecture for protocol-based reasoning and that can support the fundamental task of detecting temporal abstractions. We have developed this software module by coupling two existing systems, RÉSUMÉ and Chronus. These two systems provide complementary temporal-abstraction techniques at the application and the database levels; their encapsulation into a mediator system can, consequently, resolve the temporal-abstraction queries of protocol planners with domain-specific knowledge for the temporal-abstraction task and primary data stored in clinical databases. We show that other temporal-abstraction methods do not scale up to the data- and knowledge-intensive environments of protocol-based decision-support systems.

1 The Temporal-Abstraction Task for Protocol-Based Decision Support

Many health-care institutions would like to bring applicable clinical protocols to the attention of providers via decision-support systems that automatically examine patient data stored in legacy databases. When protocol designers represent the knowledge of eligibility criteria and of treatment plans within such systems, the system can query the database to determine specific protocols for which patients may be eligible and any interventions that are expected when a patient is seen. Institutions that have provided these types of decision-support systems have shown that even simple situation-specific advice can increase the usage of preventive care [1] and can identify adverse drug reactions [2].

Since the planning of medical therapies is highly time dependent, decision-support systems that can reason effectively about clinical protocols require a wide range of temporal deductions. In particular, an essential task in predicating protocol advice is the detection of clinically relevant **temporal abstractions** from patient data. For example, a decision-support system for the protocol-based care of patients who have insulin-dependent diabetes might notify a physician if *a significant number of a patient's preprandial (pre-meal) glucose values in the past two weeks were out of the normal range*. Such a clinical condition would alert the health-care provider that either the insulin or the meal regimen should be altered to bring the patient's glucose levels within an acceptable level.

If we analyze the condition, we notice that carrying out this temporal-abstraction task actually requires several subtasks. For example, the interpretation of glucose values depends on finding in which context ("preprandial" or "postprandial") the measurement occurred; the inference of glucose measurements into abstract states (such as "low" or "high") requires the use of clinically defined thresholds; and the aggregation of glucose state abstractions (such as the count of abnormal results) during certain intervals (such as within the past two weeks) is based on temporal-pattern matching.

Database-query languages do not currently have the ability to satisfy all these subtasks of temporal abstractions on electronically stored patient data. To overcome this obstacle, most developers of protocol-based decision-support programs have either (1) extended the data-abstraction capabilities of databases or (2) provided data-management techniques within the knowledge-based system. In this paper, we indicate that implementing temporal-abstraction methods with either of these approaches alone does not permit the seamless integration of protocol-based decision-support systems with legacy databases. We discuss two systems — RÉSUMÉ and Chronus — that support complementary aspects of the temporal-abstractions task at the application and the database levels, respectively. We show that the coupling of these systems, however, requires the system developer to specify the coordination of the two systems. To resolve this problem, we describe a novel software module (based on mediator technology) that integrates logically the temporal-abstraction mechanisms of RÉSUMÉ and Chronus.

2 Database and Knowledge-Based Approaches to the Temporal-Abstraction Task

Database-management and knowledge-based systems both support the goal of processing data. Yet, their perspectives on data processing have historically been divergent [3]. Database-management systems have been concerned primarily with giving multiple users access to large sets of consistent, permanent data on secondary-storage devices, whereas knowledge-based systems have provided a single application the ability to derive logical consequences from a comparatively smaller number of memory-resident facts. Database-management systems can serve as the central data repository of clinical information system, but they do not provide applications utilities for extracting information not explicitly stored in the database. The developers of knowledge-based systems, on the other hand, have traditionally designed decision-support programs on isolated, single-user machines; the underlying software typically does not support reliable data storage.

Because database-management and knowledge-based systems provide different, yet complementary, aspects of clinical data processing, the integration of these types of systems should be a prerequisite for a decision-support architecture that can query temporal abstractions from data in clinical databases. Most computer-based methods for temporal abstraction, however, do not support such an architecture. Some temporal-abstraction methods (such as VM [4] and TrenDx [5]) allow neither higher-level applications (e.g., protocol planner) to query results nor lower-level data sources (e.g., databases) to provide input.

Several developers of protocol planners have attempted to extend existing database-management systems with inference capabilities for temporal abstractions. This approach ensures that the output of any data abstraction can be stored back in the database in a manner consistent with primary data; the programming facilities of database-management systems, however, do not support the complex reasoning methods required by the temporal-abstraction task. For example, a database-management system that incorporates the Arden Syntax [6] — a procedural method for supporting clinical algorithms in a variety of databases — can alert a health-care provider about the occurrence of a simple temporal condition (such as a significantly low glucose value). Although this abstraction result can be placed into the central database, the expressiveness of the Arden Syntax limits its ability to provide a protocol planner more complex subtasks of temporal abstractions (such as finding the number of abnormal glucose results in the last two week period).

Another approach to the implementation of the temporal abstraction task is to incorporate both data-management and temporal-inference techniques within the knowledge-based system that performs the protocol planning. This type of architecture ensures that knowledge and data are readily available to the temporal-abstraction method from a reliable, consistent source. Such an approach does not, however, permit the temporal-abstraction method to make dynamic queries to existing databases in legacy systems. In the ONCOCIN system [7], for example, the problem-solving method for chemotherapy planning, the temporal-abstraction program for determining a patient's reaction to past chemotherapies, and the data structures for the time-stamped clinical data were all written in LISP code. In this decision-support system, the temporal-abstraction method was entirely dependent on the user's entry of data into the internal data structures. To overcome the limitations of approaches that use exclusively database or knowledge-based methods, we have instead developed generic methods that can use domain-specific temporal-abstraction knowledge and that can provide general access to temporal data stored in clinical databases.

3 The RÉSUMÉ and the Chronus Temporal-Abstraction Modules

In designing the T-HELPER system [8] — a data-management and advice system for protocol-based care of patients with HIV disease — we have attempted to avoid the problems of previous approaches by creating modular temporal-abstraction components. We, consequently, have used emerging industry-wide standards (such as UNIX, C, and SQL) as the basis of our system. To separate the domain knowledge of a protocol planner from the data-access methods of underlying database-management systems, we have also have created a pair of temporal-abstraction modules, the **RÉSUMÉ** system [9] and the **Chronus** system [10], which we have developed in the CLIPS production-rule system and the DB-Library interface to Sybase, respectively.

RÉSUMÉ and Chronus provide a complementary type of temporal deduction over patient data. RÉSUMÉ uses protocol-specific knowledge to extract from primitive data (in its fact base) high-level summaries of a patient's condition over time (such as the inference of glucose values into "low" states), whereas Chronus provides a general SQL-based data-access language to make complex temporal queries on data stored in relational databases (such as the aggregation of values within a defined time period). RÉSUMÉ, unlike Chronus, does not support queries over multiple patients or over disjoint intervals; Chronus, unlike RÉSUMÉ, does not support the identification of intervals that are not stored explicitly in the database. With the complementary action of these systems, we can support at the application or the database levels the temporal-abstraction subtasks that are the most appropriate for that level.

The flexibility of our dual systems, however, imposes a constraint on the developer of the protocol planner: She is responsible for specifying the coordination of the systems for each query. For example, to query the count of abnormal glucose-state abstractions during the past two weeks, the developer must first specify the loading of time-stamped glucose values from the database (via Chronus) into the memory-resident fact base of the RÉSUMÉ system. This loading requires a set of *mapping rules* to translate data between the database schema and the fact-base representation. The RÉSUMÉ system then creates glucose-state abstractions in the clinical context specified by the protocol planner (such as the clinical management of patients who have insulin-dependent diabetes), and the results are saved into the database. Using the query language of the Chronus system, the developer must finally specify a temporal query that determines the count of abnormal glucose-state abstractions that are stored in the database. Because this integration methods requires manual coordination, the system developer must define for each temporal-abstraction condition the procedural knowledge necessary to implement the temporal-abstraction task with the RÉSUMÉ and the Chronus systems.

4 The Tzolkin Temporal-Abstraction Mediator

To remove the need for manual coordination, we are developing a single system, called **Tzolkin**,† that can process automatically queries from different protocol-planning applications and that can make temporal abstractions as needed. Such a system is termed a *mediator*, because it serves as a middle layer between the user-oriented processing of applications and the data-manipulation methods of database systems [11]. A distinguishing feature of the mediator approach to integration is its ability to use encoded knowledge about data to create more abstract information for higher-layer applications. In Section 4.1, we describe the types of encoded knowledge required by the Tzolkin mediator to perform the temporal-abstraction task, and, in Section 4.2, we discuss the query language that is the interface to the mediator and the query-evaluation strategy that specifies the

† Tzolkin is the Mayan term for the Sun Stone, which served as an accurate representation of calendar time.

coordination between the RÉSUMÉ and the Chronus components.

4.1 The Knowledge-Based Method

In the design of a temporal-abstraction mediator, we have developed a knowledge-based method that decomposes the temporal-abstraction task into five specific subtasks: (1) temporal-context restriction; (2) vertical temporal inference; (3) horizontal temporal inference; (4) temporal interpolation; and (5) temporal-pattern matching. The first four subtasks are supported by four corresponding problem-solving mechanisms in the RÉSUMÉ program, whereas the last subtask is provided by the Chronus program. To implement these subtasks in a domain-independent manner, we have defined explicitly the knowledge requirements needed by the mechanisms; using the domain model in the protocol planner or knowledge from a domain expert, a system designer can instantiate the knowledge requirements in the knowledge-base representation (known as an *ontology*). In addition to these ontologies, the knowledge base of the mediator contains mapping rules (as described in Section 3). By unifying the mechanisms of the RÉSUMÉ and Chronus programs into a single temporal-abstraction method, we can ensure consistency and compatibility of the temporal-abstraction knowledge that is used by both components.

4.2 The Query-Evaluation Strategy

All interactions between the protocol planner, the temporal-abstraction mediator, and the clinical database occurs through message passing of queries and data. For the query language of the clinical database, we assume an SQL interface. For the query language of Tzolkin, we have made extensions to the TimeLine SQL (TLSQL) language of Chronus [12] to create **SQLA** (SQL for Abstractions). The following SQLA statement, for example, specifies the example temporal-abstraction condition in Section 1 for an individual with patient identification 1015:

```
IN CONTEXT   diabetes_management
SELECT       COUNT(parameter)
FROM         state_abs_table
WHERE        parameter =
             "glucose_state_DM_preprandial"
             AND value != "normal"
             AND patient_id = 1015
WHEN         [start_time, stop_time] AFTER
             ADDGRANULE(-14 days, GETDATE())
```

As with TLSQL, SQLA adds temporal extensions (such as the `IN CONTEXT` and `WHEN` clauses) to the standard SQL syntax. The semantics of the SQLA language, however, are defined not by the relational model but by the knowledge-based temporal-abstraction method.

To implement such queries in Tzolkin, we have defined a **query-evaluation mechanism** that uses the ontology in the knowledge base to determine the domain-specific elements of a query that are needed to process the query. Depending on which temporal-abstraction mechanisms are needed to implement the query, the query-evaluation mechanism also finds the most appropriate query-evaluation strategy. The query-evaluation mechanism inputs the procedural knowledge of the strategy into a **system-control structure** that coordinates the actions of the RÉSUMÉ and the Chronus components. We have analyzed the integration methods currently required by our dual temporal-abstraction systems, and have identified several evaluation strategies that are needed for the mediator.

The example SQLA query requires a strategy that interweaves of the temporal-abstractions mechanisms of the RÉSUMÉ and the Chronus components as follows:

1. Chronus loads data from the clinical database into the fact base of the RÉSUMÉ component (the knowledge base provides information about the scope of the primitive data required for temporal abstractions).
2. RÉSUMÉ restricts its deduction capabilities to the clinical context specified in the query, and undertakes the required mechanisms for temporal inference and temporal interpolation on the data in the fact base. (The evaluation strategy does not fix the order of the mechanisms in RÉSUMÉ, since they iterate alternately on the data and intermediate results.)
3. The Chronus component transfers the results from the fact base to temporary tables in the clinical database, and then performs SQL queries for aggregation or for complex temporal-pattern matching.

In essence, the system-control structure uses the query-evaluation strategy to automate the coordination that was previously undertaken by the system developer.

5 Discussion

In this paper, we have described a mediator system that protocol planners can query to identify automatically time-related abstractions from primary data in legacy databases. Our novel software module is based on a formal knowledge-based method that decomposes the temporal-abstraction task into five subtasks, each of which is implemented by a specific mechanism in either the RÉSUMÉ or the Chronus components of the mediator. By encapsulating these two components into a single system that can mediate queries from the protocol planner to the clinical database, we have avoided the problems of most previous approaches, which either supported complex temporal deductions within the database system or provided data-access techniques within the protocol-planner.

In contrast to most systems for temporal abstraction, the M-HTP system [13] does provide separate data-access and temporal-deduction components within a decision-support architecture. Unlike our approach, however, the database-access method is supported within an interface to the database-management system, and the temporal-abstraction method is part of the knowledge-based protocol planner. When the M-HTP system acquires patient data from the external database, the database-interface must first translate data from the database schema to the representation in the reasoning methods. Then, the protocol planner performs temporal-abstractions on the data, and stores the results internally in a temporal network.

In contrast to the temporal-abstraction component of the M-HTP system, our temporal-abstraction mediator is transferable to different decision-support systems. In our system, for example, we need only specify the mapping rules to translate data between the data representations of planner and of the database. In the M-HTP system, this mapping information is internalized by the database interface, and is not transparent to the temporal-abstraction method. Our temporal-abstraction mechanisms are also domain independent, and require only encoded domain-specific knowledge to implement the temporal-abstraction task for the protocol planner. The knowledge for creating temporal abstractions in the M-HTP system, on the other hand, is not separate from the knowledge needed for protocol planning in the domain of monitoring heart patients. In our system, the developers of protocol planners do not need to specify the technical and administrative knowledge necessary to reimplement the temporal-abstraction task, because the query-evaluation mechanism and the mapping rules generate automatically this information. In contrast, to reimplement in another architecture the temporal-abstraction component of the M-HTP system might require the system developer to change the internal codes of the database interface and of the protocol planner.

Because the mediator approach to the temporal-abstraction method is novel, the approach raises new issues for programs that interpret clinical data. For example, the mediator's evaluation of queries is a hybrid method of rule-based and database algorithms for pattern-matching languages (from the CLIPS expert-system shell and the Sybase database system); thus, we can not easily determine a general time complexity for the Tzolkin system. In the generation of a query-evaluation strategy, however, we recognize that we can optimize certain complex queries (such as temporal abstractions from multiple patients) by concurrently processing data at the mediator and database level.

Our approach also raises the issue of the defeasibility of the temporal-abstractions that we create. The current RÉSUMÉ program uses a truth-maintenance system that can permit all temporal abstractions to be withdrawn from the fact base as the result of new, contradictory data; however, neither our previous dual-system architecture nor our current mediator architecture ensures that data entered into the database is similarly entered into the fact base. To avoid this problem of nonmonotonicity, we can make all abstraction results to the protocol planner contingent on the content of the clinical database at the time the query is evaluated by the mediator. Data that are deleted or added to the database after temporal reasoning has begun will not affect the results of the abstraction process, and the results will not be saved between queries.

We are currently implementing the Tzolkin system, and still need to investigate the most appropriate solutions to the issues of optimization and nonmonotonicity. In this paper, however, we have established the necessity of such a general software module, and we have indicated the components that are sufficient to provide the functionality of a temporal-abstraction mediator.

Acknowledgments

We thank Dr. Frederic Kraemer for his example temporal-abstraction queries. This work has been supported in part by grant HS06330 from the Agency for Health Policy and Research, and by grants LM05208 and LM 07033 from the National Library of Medicine. Dr. Musen is a recipient of an NSF Young Investigator Award.

References

1. McDonald, C.J., Hui, S.L., Smith, D.M., Tierney, W.M., Cohen, S.J., Weinberger, M., and McCabe, G.P. Reminders to physicians from an introspective medical record. *Annals of Internal Medicine* 100:130–138, 1984.
2. Classen, D.C., Pestotnik, S.L., Evans, R.S., and Burke, J.P. Computerized surveillance of adverse drug events in hospital patients. *Journal of the American Medical Association* 266:2847–2851, 1991.
3. Rundensteiner, E.A. The role of AI in databases versus the role of database theory in AI: An opinion. In Meersman, R.A., Shi, Z., and Kung, C. (eds), *Artificial Intelligence in Databases and Information Systems*. North-Holland: Amsterdam, 1990.
4. Fagan, L.M. *Representing Time-Dependent Relations in a Clinical Setting*. Ph.D. Thesis, Computer Science Department, Stanford University, Stanford, CA, 1980.
5. Haimovitz, I.J., and Kohane, I.S. Automated trend detection with alternate temporal hypotheses. *Thirteenth International Joint Conference on Artificial Intelligence*. Chambrey, France. R. Bajcsy (ed), Morgan Kaufman. 1993, pp. 146–151.
6. Hripcsak, G., Clayton, P.D., Pryor, T.A., Haug, P., Wigertz, O.B., and van der Lei, J. The Arden syntax for medical logic modules. *Fourteenth Annual Symposium on Computer Applications in Medical Care*. Washington, DC. R.A. Miller (ed), IEEE Computer Society Press. 1990, pp. 200–204.
7. Kahn, M.G., Fagan, L.M., and Tu, S. Extensions to the Time-Oriented Database model to support temporal reasoning in medical expert systems. *Methods of Information in Medicine* 30:4–14, 1991.
8. Musen, M.A., Carlson, C.W., Fagan, L.M., Deresinski, S.C., and Shortliffe, E.H. T-HELPER: Automated support for community-based clinical research. *Sixteenth Annual Symposium on Computer Applications in Medical Care*. Baltimore, MD. M.E. Frisse (ed), McGraw-Hill. 1992, pp. 719–723.
9. Shahar, Y., and Musen, M.A. RÉSUMÉ: A temporal-abstraction system for patient monitoring. *Computers and Biomedical Research* 26:255–273, 1992.
10. Das, A.K., and Musen, M.A. A temporal query system for protocol-directed decision support. *Methods of Information in Medicine* (in press).
11. Wiederhold, G. Mediators in the architecture of future information systems. *IEEE Computer* 25:38–50, 1992.
12. Das, A.K., Tu, S.W., Purcell, G.P., and Musen, M.A. An extended SQL for temporal data management in clinical decision-support systems. *Sixteenth Annual Symposium on Computer Applications in Medical Care*. Baltimore, M.D. M.E. Frisse (ed), McGraw-Hill. 1992, pp. 128–132.
13. Larizza, C., Moglia, A., and Stefanelli, M. M-HTP: A system for monitoring heart transplant patients. *Artificial Intelligence in Medicine* 4:111–126, 1992.