From Guideline Modeling to Guideline Execution: Defining Guideline-Based Decision-Support Services

Samson W. Tu, M.S., Mark A. Musen, M.D. Ph.D.
Stanford Medical Informatics, Stanford University School of Medicine
Stanford, CA 94305-5479, USA

We describe the task-based approach we have taken to define guideline-based decision-support services that the EON system provides. We categorize uses of guidelines in patient-specific decision support into a set of generic tasks—decision-making, specification of work to be performed, interpretation of data, setting goal, and issuance of alert and reminders—that can be solved using various techniques. Our model contains modeling constructs required for representing knowledge used by these techniques. These constructs form a toolkit from which modeling solutions for guideline tasks can be selected. Based on the tasks and the guideline model, we define a guideline-execution architecture and a model of interactions between a decision-support server and clients that invoke services provided by the server. These services use generic interfaces derived from guideline tasks and their associated modeling constructs. We describe two implementations of these decision-support services and discuss how this work can be generalized. We argue that a well-defined specification of guideline-based decision-support services will facilitate sharing of tools that implement computable clinical guidelines.

INTRODUCTION

In recent years, guidelines and protocols have gained support as the vehicles for promulgating best practices in clinical medicine, and many researchers have proposed frameworks for modeling them in a computer-interpretable format.[1-5] However, how computer systems use computer-interpretable guidelines has not been extensively studied. This paper presents a task-based definition of a set of decision-support services that can be provided by a computer-interpretable representation of clinical guidelines.

Clarifying the services that a guideline representation can support is important for several reasons. First, the appropriate guideline representation for an application depends on the goal of the application. Some projects, such as PRESTIGE[2] and EON,[5] use computer-interpretable guidelines to provide patient-specific decision support for chronic-disease and clinical-trial therapy planning. In these uses, criteria that test for specific patient situations are paramount. Other projects, however, have different goals. Fridsma and Thomsen,[1] for example, studied communication and coordination problems involved in implementing clinical protocols in an organization. It used an information-processing approach that abstracted away specifics of individual patients that are important for patient-specific decision support. A third work modeled reporting and meta-analysis requirements of clinical-trial results.[6] 38 out of 158 classes in one version of the model were identical or very similar to those of an early version of EON. The other 120 classes modeled concepts of population, statistics and outcomes that were necessary for reasoning about trial results. These examples illustrate the need for systematic study of the relationship between application tasks and the representation requirements they place on guideline models.

Specifying services that guideline and protocol systems provide to applications is also important because of the need to share, not only guideline representation, but also the execution engines that interprets them. Building and testing execution engines is laborious and tedious, and presents a formidable obstacle to using any guideline representation formalism. If a guideline representation comes with a model of its usage and an execution engine that implements the services it supports, its evaluation and use will be much easier.

Finally, a well-defined specification of guideline-based services facilitates integrating a decision-support system into host systems that include existing electronic medical record systems. It clarifies the functionalities of the decision-support system and describes the rules for invoking these services. A generic interface specification makes possible the creation of a guideline-based decision-support system that provides a distributed service available to multiple applications.

This paper describes the *Dhammapadda* guideline modeling and execution system and the generic decision-support services interface we have developed as part of the EON project.[*]

GUIDELINE TASKS AND GUIDELINE MODEL

We abstracted a set of generic guideline tasks from a

[*] We named the EON guideline model the *Dhamma* model, after the Buddhist term for divine law. We named the guideline execution module the *Padda* server, after the Pali term for “method” or “way.” *Dhammapadda* itself is a famous work written by the historical Buddha.

variety of clinical guidelines: decision-making, specification of tasks to be performed, interpretation of data, setting goal, and issuance of alert and reminders. These tasks have been used to analyze breast-cancer clinical-trial protocols, hypertension and asthma management guidelines, an influenza-vaccination guideline, and an HIV management guideline.

Decision Making Guidelines, defined by Institute of Medicine as “statements to assist practitioner and patient decisions about appropriate health care for specific circumstances,”[7] focus on this task. In the Dhamma guideline model, we model a relevant patient state by a *scenario* that has a textual description and a precondition specifying requirements for a patient to be in this scenario. We have implemented simple if-then-else constructs and a form of argumentation—*rule-in* and *rule-out* criteria as a way of setting qualitative preferences—for decision-making (Figure 1). If a rule-out condition evaluates to true, an alternative is rejected. If the rule-out condition does not apply and a rule-in condition evaluates to true, the alternative is marked as preferred. We envision incorporating a third method for decision-making: use of expected values computed from a decision-theoretic model.

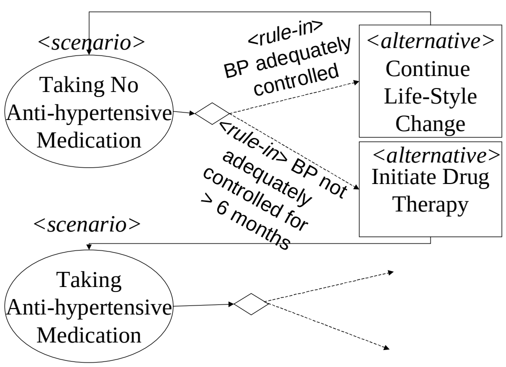

Figure 1. Example of guideline decision. Note that a guideline may not specify a preferred action in all possible circumstances. In this example, if blood pressure is inadequately controlled for less than six months, the guideline leaves the decision to the clinician’s judgment.

Setting Goals As part of the decision-making process, guidelines may set goals (e.g., “for patients with diabetes mellitus, reduce arterial blood pressure to below 130/85”). Shahar *et al*.’s Asbru model[4] offers the most sophisticated approach for modeling guideline goals and intentions. However, in the Dhammapadda model, we take a toolkit approach that allows the definition of simple goals without necessarily incorporating all of Asbru’s assumptions about temporal data. For example, for modeling a hypertension guideline, we use a relatively simple criteria language to define conditional goals such as blood-pressure targets. The criteria language allows constructs such as *last blood pressure in past 3 weeks*, references to classification hierarchies, and uses *and*, *or*, and *not* operators to construct boolean formulas.

Work Specification In contrast to guidelines that assist clinicians and patients in decision-making, clinical trial protocols emphasize work management in patient care. These protocols specify the temporal ordering, including sequence and repetition of interventions in a clinical trial, enumerate the studies that must be done regularly, and define algorithms for adjusting interventions based on reactions to previous interventions. To specify these algorithmic relationships, we created a flowchart language that includes criteria for defining temporal constraints among tasks and relationships such as *followed by*, *repeat every so often until*, *do N times*, and *do concurrently* to represent sequencing, repetition, and concurrency information (Figure 2).

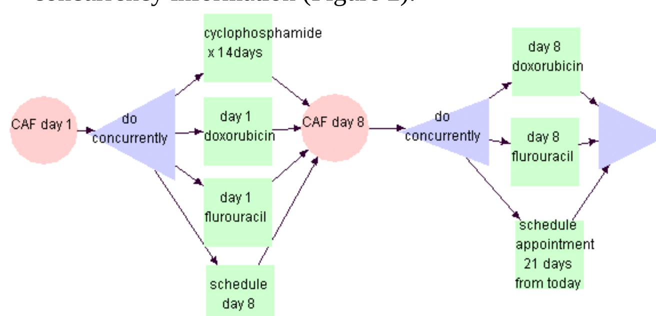

Figure 2. Example of concurrent actions. On Day 1, cyclophosphamide, doxorubin, and fluorouracil should be prescribed. On Day 8, only doxorubin and fluorouracil should be prescribed. Clinician should also schedule the next encounter on each clinic day. Arrows represent the “followed by” relationship.

Interpretation of Data Finally, data interpretation is part of every clinical decision. The Dhamma guideline model has a variety of techniques for this task. Abstract concepts can be derived from concrete entities through a classification hierarchy (*e.g.*, *angina* is a kind of *cardiovascular disease*). The presence or absence of a clinical syndrome can be derived from a boolean combination of criteria that check for specific patient conditions (*e.g.*, *isolated systolic hypertension* is presence of hypertension with high systolic blood pressure but normal diastolic pressure). Measures like risk factors for cardiac events can be computed from equations with estimated parameters. Interval-based abstractions, such as toxicity episodes defined in cancer clinical trials, can be derived from time-stamped data through the use of temporal abstractions that map ranges of laboratory-test results or signs and symptoms to toxicity grades.[8]

EXECUTION ARCHITECTURE

In the EON project, a guideline modeler uses the Protégé-2000 knowledge-editing environment[9] to create and maintain clinical guidelines and protocols using the modeling constructs in the generic Dhamma guideline model. Part of the modeling process involves creating or adapting a *domain ontology* that defines terms and relationships that are relevant to the medical specialty. To model the hypertension guideline, for example, we created classification hierarchies that include definitions of the co-morbidities of hypertension and classes of anti-hypertensive drugs. The hypertension domain ontology provides a data model and a controlled vocabulary for referencing patient conditions and therapies that are relevant to managing hypertension. Together with the Dhamma guideline model, it allows the guideline author to represent clinical guidelines in a structured and computer-interpretable format.

To provide patient-specific guideline-based decision support, we created the *Padda Guideline Execution Server* (or *Padda Server*). The Padda Server applies a clinical guideline to patient data queried from a host system's database and generates advisories for Padda clients (Figure 3). Within the Padda Server, a knowledge-base handler manages access to the guideline knowledge base and the domain ontology via the application-programming interface provided by Protégé-2000. A data mediator performs two crucial functions: (1) it maps the patient data model assumed by the guideline model to the data model of the host system, and (2) it maps terminology in the domain ontology, such as names of laboratory test results, to the terminology used in the host database system. The data-handler component implements each type of query specified in the Dhamma guideline model.

For a specific guideline and patient, the Padda Server must determine if the guideline is applicable to the patient, and subsequently, implement a model of interaction with its software clients. The interaction model is based on (1) the guideline tasks described in the previous section and (2) the concept that a decision-support system and a clinician are engaged in cooperative problem-solving, where the clinician can override any conclusion made by the decision-support system. Thus, when the Padda Server uses patient data to determine if a patient-specific guideline goal has been achieved, an external user can require it to assume that that goal has been achieved. Similarly, the Padda Server uses patient data to suggest that a patient is in a specific scenario, and that, as a result, tasks such as laboratory tests should be performed. The server may also suggest that certain alternatives at a decision point are preferred. In each case, the user may override the system's conclusions. Table 1 enumerates guideline tasks, Padda server services, and possible user actions.

GUIDELINE-SERVICE INTERFACE

The guideline services and the client-server interaction model described in the previous section give rise to a specification of the interface between the Padda Server and its client. The specification, written in Common Object Request Broker Architecture Interface Definition Language (CORBA IDL), consists of two parts: (1) the methods with which client and server interact with each other, and (2) description of the data structures that are passed between the server and clients. Figure 4 shows a small fragment of the interface specification.

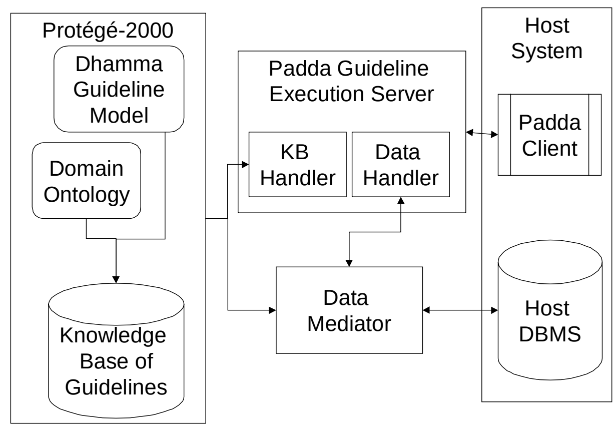

Figure 3. Execution Architecture of the EON System. The Padda Guideline Execution Server loads and queries a guideline knowledge base implemented in Protégé-2000 and uses a data mediator to access and map patient data stored in a host system.

Table 1. Guideline services provided by the Padda Guideline Execution Server.

Guideline Task	Padda Services	User Action
Goal setting	Determine patient-specific guideline goal and if it has been achieved	Specify whether guideline goal has been achieved
Decision making	Enumerate current decision points and preferred alternatives	Specify alternatives to be chosen
work specification	Enumerate tasks to be performed according to guideline and user decisions	Specify tasks to be performed
Data interpretation	Present conclusions derived from data	Specify data to use

As the Padda Server applies a guideline to a specfic patient situation, it traverses the clinical algorithm associated with the guideline, and determines the appropriate guideline scenario for the patient. It may discover that a management decision must be made. It creates an instance of *Guideline_Action_Decision* as displayed in Figure 4 and fills in the *guideline_id*, *current_location*, and *selection_method* slots from the guideline representation. The Padda Server fills the *action_choices* slot by finding the possible action choices at that point and evaluates the rule-in and rule-out criteria assoicated with each choice. For each action choice, it creates an *Action_To_Choose* record. The result of evaluating the rule-in and rule-out criteria is stored in a *Justication* record and a *preference* such as *preferred*, *neutral*, or *rule-out* is computed for that choice. The Justification record refers to the evaluated guideline criteria, the patient data used, and assumptions, if any, made in evaluating the criteria. The *action_specifications* slot defines the set of actions to be performed if the choice is selected.

```
 struct Action_To_Choose {
   string name;
   Preference preference;
   Justification  preference_justification;
   sequence<Action_Spec_Record>
action_specifications;
   Guideline_Entity action_step;
   string description;
 };
 struct Guideline_Action_Decision {
   string guideline_id;
   Guideline_Entity current_location;
   sequence<Action_To_Choose> action_choices;
   SelectionAlternatives selection_method; }
```

Figure 4. A fragment of the Padda guideline service interface specification.

The interaction protocol between the Padda Server and its clients is specified as a set of methods associated with the Padda Server's IDL interface. A Padda client invokes these methods to select the guideline and the patient for the current Padda session, to determine the current location of the patient within the guideline's clinical algorithm, to receive recommended tasks and guideline preferences on alternatives for the current decision. In each case, the interface allows the clinician to specify the current patient scenario, alternatives to be chosen, and tasks to peformed.

RESULTS

A prototype of this decision-support service interface was implemented for the PRODIGY system in the United Kingdom.[10] The prototype modeled a guideline for managing adult asthma. It used a web-based user interface and assumed that a clinical user interacts with the decision-support system in a dialog (Figure 5a) where the system presents to the user management alternatives for a given scenario, tasks to perform, and data to obtain. The user enters pertinent data and makes decisions regarding patient-management choices.

A second version of this decision-support interface was implemented in the ATHENA project in collaboration with the Veteran's Affairs Palo Alto Clinic.[11] This project developed a guideline decision-support system for managing hypertension. In the ATHENA implementation, guideline-based advisories are displayed in a window when a clinician selects a patient from the electronic medical record. No interaction between the decision-support system and the clinician is assumed. We supported this style of displaying guideline advisories by interposing a local Padda session mediator that implements a simplified decision-support service interface (Figure 5b). Upon receiving a request to generate advisories for a patient, the session mediator, on behalf of this client, carries out a dialog with the Padda Server until it generates advisories to be displayed to the user. Within the session mediator, we encoded a set of local policies that allow the mediator to make decisions that the Padda Server must otherwise obtain from an external user. The policies include (1) selecting the scenario computed by the Padda Server, (2) selecting and expanding preferred patient management choices, and (3) in the absence of current blood pressure information in the database, making alternative assumptions about the status of a patient's blood pressure.

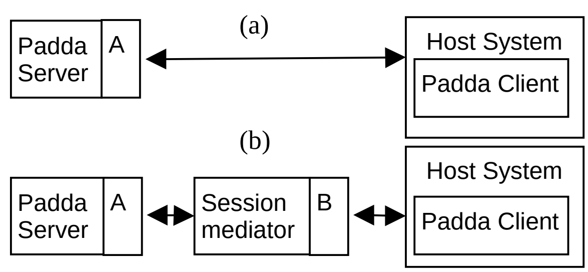

Figure 5. Alternative interfaces to the Padda Server. (a) The host system and the client directly use interface A provided by the Padda Server. (b) An intermediary provides an alternative interface to the client while using A to communicate with the Padda Server.

DISCUSSION

Medical informatics has produced decision-support technologies that showed great promise. However, few systems have achieved wide-spread and sustained use. In the area of guideline-based applications, many reasons contribute to this problem. They include the heterogeneity of terminologies and clinical information systems, and difficulties in achieving consensus on content as well as models. This paper addresses one issue that is needed to allow sharing of guidelines for specific uses: specifying guideline-based decision-support services.

We identified a set of tasks—decision-making, work specification, goal setting, data interpretation, and sending alert and reminders—and structured our guideline model to represent alternative solutions to first four of these tasks. We showed that a specification describing the interaction and the data structures necessary to support decision-support services follows from requirements derived from the tasks that a guideline attempts to achieve.

A comparison with Arden Syntax[3] is illuminating. Arden Syntax is designed to share medical logic modules (MLMs). MLM is a good formalism for implementing event-driven alerts and reminders not covered by EON's Dharmapadda system. MLMs have a simple interaction model between a host system and a guideline execution system: an external event, specified by the MLM's evoke slot, triggers an MLM. If the condition in the *logic slot* evaluates to true, one of three actions is taken: (1) a message is written to some destination, (2) expressions are returned to a calling MLM, or (3) an event is called to invoke other MLMs. This paper argues that performing guideline tasks in the context of computer/user interactions may require a richer interface specific to these tasks.

CORBAMed has issued a Healthcare Data Interpretation Facility Request for Proposals (RFP) that solicits "submissions of a general-purpose infrastructure capable of accommodating a variety of intelligent transforms useful for healthcare data interpretation in the context of decision support." [12] The RFP's emphasis is on a general-purpose and flexible decision-support interface that allows introspection on types of available intelligent data transform and the technique, methodology, or algorithm used. These issues complement those addressed in this paper. Instead of a generic data-transformation interface, we propose an interface that supports a specific set of guideline-based decision-support tasks. As the generic introspective interface specification format becomes available, we will be able to recast the Dhammapadda decision-support interface, currently implemented entirely in CORBA IDL, in the new formalism.

Computer-interpretable guideline representation can provide services that cannot be supported using text-based format. A specification of these services, by describing ways through which decision-support services are delivered to host systems, informs the modeling process and facilitates the integration of such services into host systems. By developing guideline execution engines that provide such decision-support services, not only do we permit the sharing of formally represented guidelines, we also make the use of these guidelines a shared resource.

ACKNOWLEDGMENTS

This work has been supported in part by grant LM05708 from National Library of Medicine, a grant from FastTrack Systems, Inc., and DARPA contract #N66001-94-D6052. We are grateful to Doug Fridsma, Peter Johnson, and Mor Peleg for discussions related to the modeling of clinical guidelines and to Mary Goldstein for the collaborative development of ATHENA system.

REFERENCES

1. Fridsma, D. B., –Thomsen, J. Representing Medical Protocols for Organizational Simulation: An Information Processing Approach. Computational and Mathematical Organization Theory 1998;4(1):71–95.

2. Gordon, C., Veloso, M. The PRESTIGE Project: Implementing Guidelines in Healthcare. Medical Informatics Europe '96: IOS Press, 1996:887–891.

3. Hripcsak, G., Clayton, P. D., Pryor, T. A., Haug, P., Wigertz, O. B., Van der lei, J. The Arden Syntax for Medical Logic Modules. Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care. Washington, D.C., 1990:200–204.

4. Shahar, Y., Miksch, S., Johnson, P. The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. Artificial Intelligence in Medicine 1998;14:29–51.

5. Tu, S. W., Musen, M. A. A Flexible Approach to Guideline Modeling. In: Nancy Lorenzi P, ed. 1999 AMIA Annual Symposium. Washinton D.C.: Hanley & Belfus, Inc., 1999:475-497.

6. Sim, I., Rennels, G. Developing A Clinical Trials Ontology: Comments on Domain Modeling and Ontology Reuse. Stanford, CA: Stanford Medical Informatics, 1995.

7. Field, M. J., Lohr, K. N., eds. Clinical Practice Guidelines: Directions for a New Program. Washington, D.C.: National Academy Press, 1990.

8. Shahar, Y. A Framework for Knowledge-Based Temporal Abstraction. Artificial Intelligence 1997;90(1-2):79–133.

9. Grosso, W. E., Eriksson, H., Fergerson, R., Gennari, J. H., Tu, S. W., Musen, M. A. Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000). In: Gains BR, Kremer R, Musen M, eds. The 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop. Banff, Canada, 1999:7-4-1–7-4-36.

10. Johnson, P. D., Tu, S. W., Booth, N., Sugden, B., Purves, I. N. A Guideline Model for Chronic Disease Management in Primary Care. In: Overhage JM, ed. AMIA Annual Symposium. Los Angeles, USA, 2000:submitted.

11. Goldstein, M. K., Hoffman, B. B., Coleman, R. W., et al. Operationalizing Clinical Practice Guidelines While Taking Account of Changing Evidence: ATHENA, an Easily Modifiable Decision-Support System for Management of Hypertension in Primary Care. In: Overhage JM, ed. AMIA Annual Symposium. Los Angeles, USA, 2000:submitted.

12. OMG. Healthcare Data Interpretation RFP. http://www.omg.org/techprocess/meetings/schedule/Healthcare_Data_Interpretation_RFP.html: 1998.