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 our task-based approach to defining the 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—making of decisions, specification of work to be performed, interpretation of data, setting of goals, and issuance of alert and reminders—that can be solved using various techniques. Our model includes constructs required for representing the knowledge used by these techniques. These constructs form a toolkit from which developers can select modeling solutions for guideline task. 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 the best practices in clinical medicine, and many researchers have proposed frameworks for modeling them in a computer-interpretable format. However, how computer systems use computer-interpretable guidelines has not been studied extensively. 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.

Providing a clear description of 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 and EON, 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, in contrast, have different goals. Fridsma and Thomsen, for example, studied the communication and coordination problems involved in implementing clinical protocols in an organization. They used an information-processing approach that abstracted away the specific data on individual patients that are important for patient-specific decision support. A third work modeled reporting and meta-analysis requirements of clinical-trial results. The other 120 classes modeled concepts of population, statistics, and outcomes that were necessary for reasoning about trial results, but were not needed for patient-specific decision support. These examples illustrate the need for systematic study of the relationship between application tasks and the representation requirements that these tasks 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 representations, but also the execution engines that interpret them. Building and testing execution engines are laborious and tedious tasks that together present a formidable obstacle to the use of any guideline representation formalism. If a guideline representation comes with a model of its usage and an execution engine that implements the services that it supports, its evaluation and use will be much easier.

Finally, a well-defined specification of guideline-based services facilitates integration of a decision-support system with 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 Dhammadaka 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 variety of clinical guidelines: making of decisions, specification of tasks to be performed, interpretation of data, setting of goals, and issuance of alert and

* We named the EON guideline model the Dharma 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." Dhammadaka itself is a famous work written by the historical Buddha.
reminders. These tasks have been used to analyze breast-cancer clinical-trial protocols, hypertension, asthma, and HIV management guidelines, and an influenza-vaccination guideline.

**Making of Decisions Guidelines**, defined by Institute of Medicine as "statements to assist practitioner and patient decisions about appropriate health care for specific circumstances,"7 focus on the decision-making task. In the Dharmapadda model, we model a relevant patient state by a *scenario*. Each such scenario has a textual description and a precondition specifying requirements for a patient to be in it. We 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.

![Figure 1. Example of a guideline decision.](image)

**Setting of 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 and colleagues’ Asbru model8 offers the most sophisticated approach for modeling guideline goals and intentions. However, in the Dharmapadda model, we take a toolkit approach that allows us to define 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 *most recent blood pressure within 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 specify the temporal ordering of interventions in the clinical experiments, 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).

**Interpretation of Data** Data interpretation is part of every clinical decision. The Dharmapadda 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

![Figure 2. Example of concurrent actions.](image)
Dharma 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. This 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 Dharma guideline model.

For a specific guideline and patient, the Padda server must determine whether the guideline is applicable to the patient, and, subsequently, must 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 assumption 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, the Padda server uses patient data to suggest that a patient is in a specific scenario, and that, as a result, tasks such as ordering specific laboratory tests should be performed. The server may also suggest that one or another alternative is preferred at a decision point. In each case, the user may override the system's conclusions. Table 1 enumerates guideline tasks, Padda server services, and possible user actions.

As the Padda server applies a guideline to a specific 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; it then evaluates the rule-in and rule-out criteria associated with each choice. For each action choice, it creates an Action_To_Choose record. The result of its evaluation of the rule-in and rule-out criteria is stored in a justification 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.

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

---

**Table 1. Guideline services provided by the Padda guideline-execution server.**

<table>
<thead>
<tr>
<th>Guideline Task</th>
<th>Padda Services</th>
<th>User Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting of goals</td>
<td>Determine what the patient-specific guideline goal is and whether it has been achieved</td>
<td>Specify whether guideline goal has been achieved</td>
</tr>
<tr>
<td>Making of decision</td>
<td>Enumerate what current decision points and preferred alternatives are</td>
<td>Specify what alternatives to choose</td>
</tr>
<tr>
<td>Specification of work</td>
<td>Enumerate what tasks to perform according to the guideline and to user decisions</td>
<td>Specify which tasks to perform</td>
</tr>
<tr>
<td>Interpretation of data</td>
<td>Present conclusions derived from data</td>
<td>Specify which data to use</td>
</tr>
</tbody>
</table>

---

**Figure 3. Execution architecture of the EON system.**

**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) 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 fragment of the interface specification.
decision. In each case, the interface allows the clinician to specify the current patient scenario, alternatives to be chosen, and tasks to be performed.

```c
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 interface specification for the Padda guideline service.

RESULTS

We implemented a prototype of this decision-support service interface for the PRODIGY system in the United Kingdom.10 This prototype modeled a guideline for managing adult asthma. It used a web-based interface and assumed that a clinical user conducts a dialog with the decision-support system (Figure 5a). During the dialog, the system presents 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. Because this prototype obtained all necessary data from the user through a web interface, it was not necessary to link the system to a host electronic medical-record system.

In collaboration with the Veteran's Affairs Palo Alto Clinic, we implemented a second version of this decision-support interface for the ATHENA project.11 In this project, we 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. We assume no interaction between the decision-support system and the clinician. 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). When it receives 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 the server 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 management choices; and (3) in the absence of blood-pressure information in the database, making alternative assumptions about the status of a patient's blood pressure.

For the ATHENA implementation, we developed a database mediator, called Athenaem,12 that maps the VA database onto the data model of the EON system. It does so in two steps: (1) transform the legacy hierarchical database into a relational database, and (2) create the temporal database required by EON and map local terminology to the guideline terminology used in the knowledge base.

The two implementations demonstrate the use of a standard decision-support service interface, and hence a shared guideline execution engine, in two significantly different application environments. However, as both implementations model guidelines for management of patients how have chronic diseases, the generalizability of the interface for the other types of guidelines and protocols remains to be tested.

![Figure 5. Alternative interfaces to the Padda server](image)

DISCUSSION

Medical informatics has produced decision-support technologies that show great promise. However, few systems have achieved widespread 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 the difficulties in achieving consensus on content as well as on models. Our work addressed one component required for sharing of guidelines for specific uses: Specification of guideline-based decision-support services.

We identified a set of tasks—making of decisions, specification of work, setting of goals, interpretation of data, and issuance of 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. Our current implementation does not deal with the situation where multiple guidelines may be applicable at the same time.
Because of the complexity of medicine, simply aggregating recommendations from two guidelines is not appropriate. Instead, guidelines should be explicitly authored for multiple co-morbidities.

A comparison with Arden Syntax is illuminating. Arden Syntax is designed to share medical logic modules (MLMs). The 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. We have argued that performing guideline tasks in the context of computer–user interactions may require a richer interface that is specific to these tasks.

CORBA Med 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.” The RFP’s emphasizes a general-purpose and flexible decision-support interface that allows introspection on types of available intelligent data transforms. These issues of generality and introspection 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. When the generic introspective interface-specification format becomes available, we will be able to re cast the Dharmapadda decision-support interface, currently implemented entirely in CORBA IDL, in the new formalism.

Computer-interpretable guideline representation can provide services not supported by a text-based format. A specification of these services, which describes the ways that 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, but also we make use of the 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-D5052. 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. Lyn Dupré and Valerie Natale edited the manuscript.

REFERENCES