An Architecture for a Distributed Guideline Server

Mike Barnes, M.D. and G. Octo Barnett, M.D.
Laboratory of Computer Science, Massachusetts General Hospital
Boston, Massachusetts

A number of barriers exist which prevent the widespread integration of practice guidelines with electronic medical record systems. These include dependencies on clinical databases, as well as problems with converting existing guideline specifications into computable rules. We are developing the MBTA (Modeling Better Treatment Advice) practice guideline system which uses a distributed client-server architecture and an object-oriented data representation to support practice guidelines usable by different electronic medical record systems. We describe the structure and organization of MBTA, focusing on how an open systems design, combined with principles of guideline implementation, can be used to provide a general purpose guideline server for use by a variety of clinical workstations.

INTRODUCTION

Clinical practice guidelines are playing an increasing role in medical practice, but their full potential for improving care has not yet been realized. One reason for this is that simple guideline dissemination, without integration into the process of care delivery, has not been shown to impact the behavior of clinicians [1, 2]. However, previous studies have demonstrated that reminders to clinicians at the time they are making clinical decisions can result in increased compliance and improved care [3–5]. Electronic medical record systems have the potential of making a great impact in the increased utilization of guidelines, since these systems are often used by clinicians at the time care is delivered. However, there are a number of barriers which must be overcome to integrate guidelines within these systems.

One barrier to implementing guidelines for clinical workstations is, as McDonald has pointed out, that some guidelines are more 'decidable' than others [6]. Decidable guidelines are easier to integrate with workstations since they tend to have well defined algorithms, standard clinical vocabularies, and use parameters typically found in a clinical database. However, since guidelines are not written with computer implementation in mind, most guidelines have components which present barriers to workstation integration. Designers who integrate existing guidelines must make assumptions about poorly specified or ambiguous aspects of the guideline [7].

Existing electronic medical record systems which support clinical alerts and reminders, such as HELP [8] and the Regenstrief Medical Record System [9], tend to be highly dependent upon the vocabulary and data structures of a particular clinical database. Because of the close coupling between the database and guideline rules, it is difficult for a given set of rules to be used by a database different from the one for which it was designed [10].

To address these problems, we are developing the MBTA (Modeling Better Treatment Advice) guideline server. MBTA uses a client-server architecture, where a client such as a clinical workstation or World Wide Web (WWW) interface program, connects to an MBTA server and enters into a series of transactions. After receiving an initial set of data from the client, the server executes a series of modules encapsulating guideline rules to produce a set of frame-like data objects. An explanation facility then uses the contents of those objects to produce a text explanation which is returned to the workstation. To illustrate the function of MBTA, we will use examples from a WWW expert system based on portions of the AHCPR Urinary Incontinence in Adults guideline [11] and a cholesterol guideline module based on the National Cholesterol Education Program (NCEP) guideline [12].

MBTA OVERVIEW

There are three basic elements which comprise the core of MBTA: object groups, modules, and explainers. *Object groups* represent MBTA's structured data model, while *modules* are the procedural elements which execute guideline logic. An example system which shows the relationship between modules and object groups is shown in Figure 1. This represents an MBTA system which implements pneumococcal and cholesterol guidelines, and interfaces to either a WWW browser or clinical workstation. In this example, either the clinical workstation or a WWW interface module creates a set of object groups (such as 'Family History') which are used as input by the modules which imple-

0195–4210/95/$5.00 © 1995 AMIA, Inc.

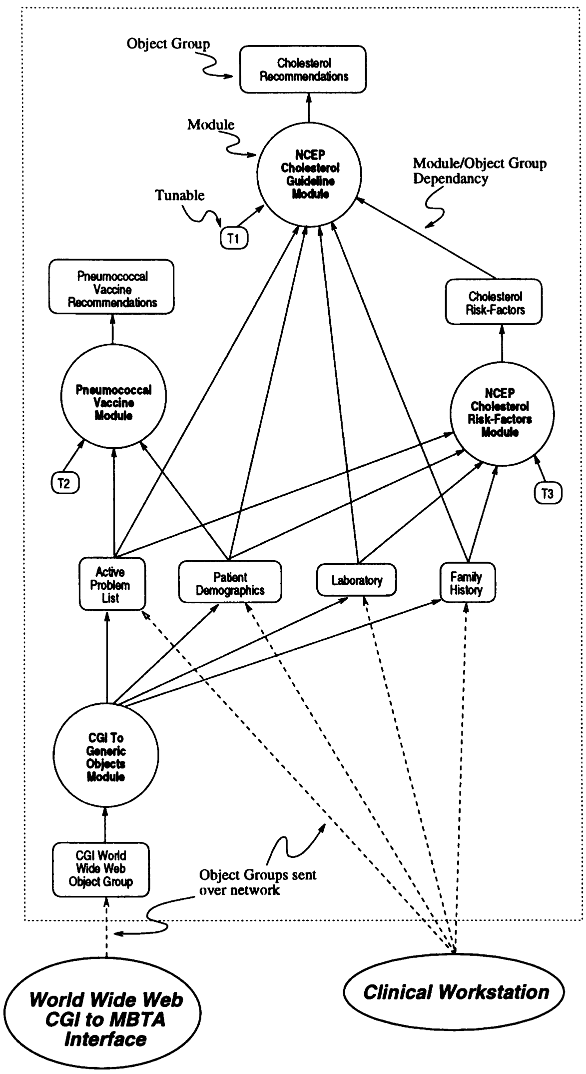

Figure 1: MBTA system showing data flow and relationships between modules and object groups.

ment the guidelines. Each guideline module uses the contents of these object groups along with guideline logic to generate one or more object groups as output. Some of the object groups created by modules (such as 'Cholesterol Risk-Factors') are used as input by other modules. The relationship between object groups and modules form a directed acyclic graph which is used to determine the order in which modules are executed. That is, modules are executed starting at the bottom of the graph, progressing upward. The primary result of this is the creation of the object groups 'Pneumococcal Vaccine Recommendations' and 'Cholesterol Recommendations.' Both of these contain structured data which is used by the last basic component: explainers.

Explainers are a special type of module which use the contents of object groups to generate prose explanations. They are run after a set of modules have been executed. This prose output is then passed back to the client for display. The overall schema is that modules are executed in a bottom-up fashion, followed by a top-down explanation. We will now describe each of the basic components in more detail.

Data Model

One of the more difficult aspects of developing a guideline server is choosing an underlying data model which is robust and adaptable to a wide variety of clinical information systems. There is no perfect model, and any choice involves a number of compromises. Our representation is based on a *prototype-instance object model* [13]. In this model, objects are frame-like collections of slot-value pairs. A simple inheritance mechanism is provided, but not necessary for most clinical data representation. In MBTA, a set of objects which have a related purpose are collected into *object groups*. Each object within an object group contains one or more slot-value pairs. Values may consist of numbers, strings, other objects, or lists of any of these. Additionally, each value, list, or item within a list, may contain a temporal value, or set of attributes. MBTA uses a simple syntax for representing the contents of object groups. For example, here is an object group generated by a cholesterol risk factor module to characterize a patient's risk factors:

```
object-group Cholesterol-Risk-Factors {
    object number-of-factors
        maximum 4
        minimum 3
        number-known 3
        two-or-more true;
    object risk-factors
        known [MALE-GE-45 DIABETES HDL-LOW]
        unknown [FAMILY-HX-CHD];
    object desired-data
        wanted-items [FAMILY-HX-CHD]
        wanted-objects [Family-History.Premature-CHD.status]; }
```

This object group contains three objects. The first, 'number-of-factors,' describes characteristics about the number of cholesterol risk factors a patient has. The actual number of possible risk factors is a range because some factors may be unknown. Therefore, both the number known, as well as the range boundaries are specified. Because the NCEP Cholesterol guideline rules concerning risk factors are based on two or more risk factors being present, this information is placed on the slot 'two-or-more.' The object 'risk-factors' contains two slots, each containing a vocabulary list of all known or unknown risk factors. The 'desired-data' object contains slots about data items which are un-

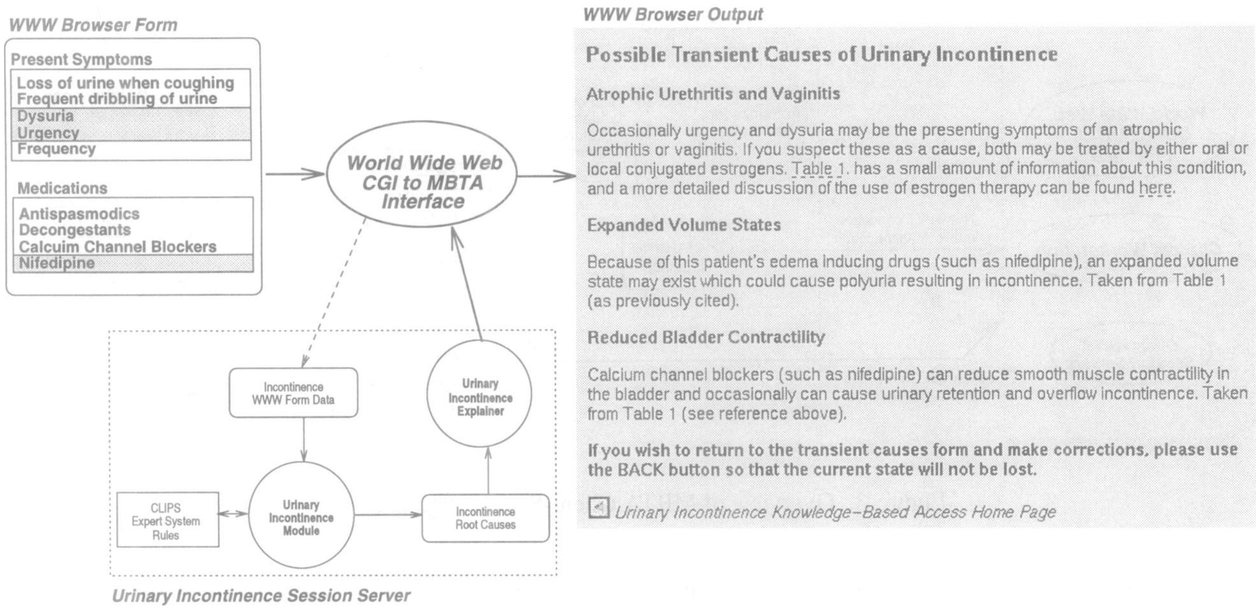

Figure 2: Overview of World Wide Web interface to Urinary Incontinence Guidelines showing explainer output.

known. These are provided primarily for feedback to a clinical workstation for possible clarification. The 'wanted-items' slots contains this information as a list of vocabulary terms, while 'wanted-objects' is a list of MBTA object names where the missing data items are normally stored.

Temporal data in MBTA is represented either as time stamps, time intervals, or durations. In MBTA object notation, temporal values are indicated by following an object value, or component of an object value (list or list member), with '<t: *temporal-value*>.' An example set of LDL cholesterol values might be represented as:

```
object-group Laboratory {
    object Cholesterol-LDL = [110<a:fasting><t:1995-03-12>
        105<a:fasting><t:1995-02-28>
        114<a:fasting><t:1995-01-28>]; }
```

Additionally, each value or list item can have a set of attributes. Each attribute consists of a name and optional value. Attributes without values denote the presence or absence of a condition, such as '<a:fasting>' to denote if a lab was obtained while the patient was fasting.

Modules

In MBTA, the object groups described above are used as input and output to a set of procedural elements, termed *modules*. There is no specific syntax for modules, the only requirements are that each module declares which object groups it uses for input and output. We have not defined a particular module syntax because we do not feel that any single syntax is expressive enough to represent the entire domain of guidelines. Rather, we created a set of interface libraries allowing various languages and programs to be called as modules from MBTA. These libraries provide a convenient method to map between MBTA object groups and the native data structures available for each particular language or program. Currently, we have implemented interface libraries to C++, the CLIPS expert system, and a variant of Common Lisp.

Explainers

Another primary component of MBTA are *explainers*. Explainers are a special type of module whose purpose is to take an object group, or collection of object groups, and produce a text explanation using the contents of those groups. Explainers are called only after all pertinent modules have been executed.

To illustrate explainers we use an example from an MBTA system we developed which allows clinicians to input patient specific history and physical findings using a WWW forms interface and receive advice relating the patient's findings to possible causes of transient urinary incontinence. The overall sequence is shown in Figure 2. In this example, the clinician selects the shaded items and submits them to a Common Gateway Interface (CGI) program which converts the Hypertext Mark-Up Language (HTML) forms data to an object group. This program then connects to an MBTA server, transmits the object group, and tells the sever to execute the 'Urinary Incontinence Module.' This module interfaces with a set of production rules which determines a set of 'root causes' of transient incontinence from the clin-

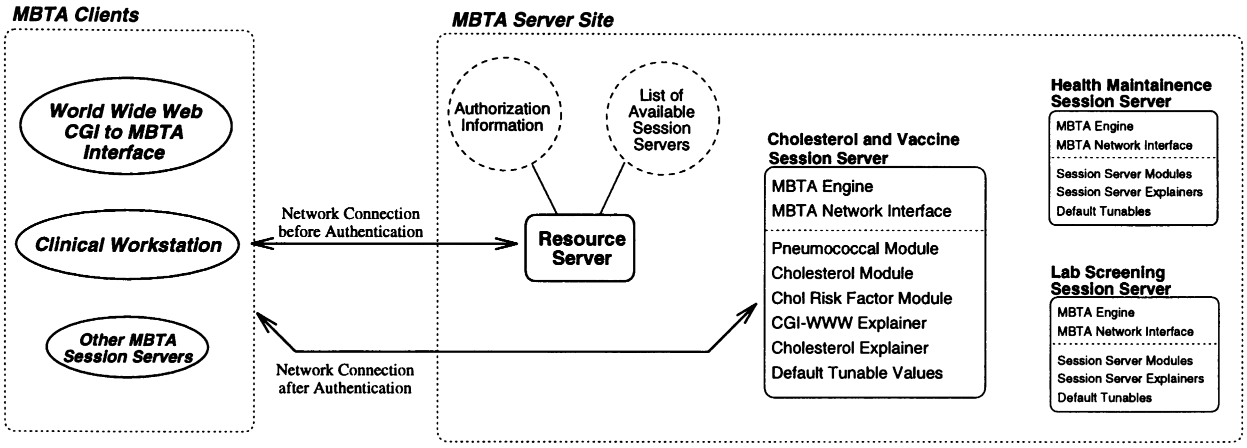

Figure 3: Overview of MBTA Client-Server Organization.

ician's input. These root causes are encapsulated in the 'Incontinence Root Causes' object group, a portion of which is shown here:

```
object-group Incontinence-Root-Causes {
  object rc-LIST
    members [rc-atrophic-urethritis-vaginitis
             rc-expanded-volume-states
             rc-reduced-bladder-contractility];
  object rc-atrophic-urethritis-vaginitis
    symptoms [sy-urgency sy-dysuria];
  object rc-expanded-volume-states
    explained-by [dc-edema-inducing];
  object rc-reduced-bladder-contractility
    explained-by [dc-ca-channel-blockers];
  object dc-edema-inducing
    has-DrugInstance [di-nifedipine];
  object di-nifedipine
    has-DrugClass [dc-edema-inducing
             dc-ca-channel-blockers]; }
```

All of the root causes are placed in the 'members' slot of the 'rc-LIST' object. Each element of this list is also the name of an object. The reason a particular root cause was selected is denoted by the slots of its associated object. For example, reduced bladder contractility is a root cause because the patient is on a drug (nifedipine) from an edema inducing drug class. In this example, 'rc-' is 'root cause,' 'dc-' is 'drug class,' 'sy-' is 'symptom,' and 'di-' is 'drug instance.'

The 'Urinary Incontinence Explainer' takes the contents of this object group and generates the HTML output which is sent back to the clinician via the CGI-MBTA interface program. The explanation model used by MBTA is a combination of canned phrases, procedural logic, and a set of general rules for combining words, phrases, and paragraphs into coherent text. Our general approach to text production is adapted from P. Miller's PROSENET system for generating polished prose using the contents of a set of frames [14].

Operational Model

MBTA follows a client-server model, whose basic features are shown in Figure 3. An MBTA client, such as a clinical workstation, makes an initial connection to a *resource server*. Typically, there is a single resource server for each MBTA site whose primary purpose is authentication. Once authentication has occurred, the client requests connection to a collection of modules, explainers, and object groups known as a *session server*. A session server is a program comprised of a group of modules that have been combined with an MBTA front end which handles communications with the client and other MBTA servers. The client then communicates with the session server by sending it commands, and receiving replies and error messages. Each MBTA client uses a simple protocol to communicate with MBTA servers.

Dealing with Guideline Ambiguity

One important problem which we have encountered when implementing guidelines is dealing with incomplete or ambiguous guideline items. For example, smoking is considered a risk factor in the cholesterol guidelines, but it is not specified how long a patient must have quit smoking for this to no longer be a factor. A similar example is how long lab values are considered valid. Our approach to this problem is to determine these ambiguous and incompletely specified items and provide a set of clarifying defaults. These defaults are isolated in a type of object group known as a *tunable*. We also determine guideline specific parameters which are candidates for customization and also place these in the tunable. The tunable for our cholesterol risk factor module (denoted as 'T3' in Figure 1) is represented as:

```
object-group Cholesterol-Risk-Factors-Tunable {
    object time-lab-is-valid = 0<t:5 years>;
    object time-from-tobacco-use = 0<t:1 year>; }
```

In this example, we have defined parameters for the length of time a lab is considered valid, and the amount of time after a patient quits smoking that smoking is still a risk factor. The initial tunable with its default values is created at the time the module is designed. Individual clinical workstations can override all or part of a tunable to adapt it for a particular environment.

RESEARCH DIRECTIONS

We believe that the MBTA architecture provides a good foundation for integrating knowledge-based guideline access with both clinical workstations and WWW browsers. Our current focus is integrating MBTA with the computer-based patient record being developed in our lab. We are deriving a clinical database to object group mapping system, as well as considering methods for implementing feedback to clinical workstations from MBTA. Since many parts of the system exist as C++ library functions, another goal is to make higher level interfaces, especially in the area of explainers. We are developing a syntax for expressing explainers, and a translator to convert this syntax into C++.

Our efforts have emphasized a number of general problems with translating current practice guidelines for use with electronic medical record systems. Because feedbacks and alerts generated by clinical information systems have been shown to be a viable method of increasing the impact of practice guidelines, it is important for groups which create guidelines to realize that their recommendations may be integrated into electronic medical record systems and design guidelines in an unambiguous fashion.

ACKNOWLEDGMENTS

This work is supported in part by NLM Grant LM5200 and AHCPR Grant HS6575. Dr. Barnes is supported by NLM training grant LM7092.

References

[1] Lomas J, Anderson GM, Karin DP, Vayda E, Enkin MW,, Hannah WJ. Do Practice Guidelines Guide Practice? The Effect of a Consensus Statement on the Practice of Physicians. *NEJM*, 321:1306–1311, 1989.

[2] Hill MN, Levine DM,, Whelton PK. Awareness, Use, and Impact of the 1984 Joint National Committee Consensus Report on High Blood Pressure. *Am J Public Health*, 78:1190–1194, 1988.

[3] Barnett GO, Winickoff R, Dorsey JL, Morgan MM,, Lurie RS. Qualtity Assurance Through Automated Monitoring and Concurrent Feedback Using a Computer-Based Medical Information System. *Med Care*, 16:962–970, 1978.

[4] Stiell IG, McKnight RD, Greenberg GH,, others . Implementation of the Ottawa Ankle Rules. *JAMA*, 271:827–832, 1994.

[5] McDonald CJ. Protocol-based Computer Reminders, the Quality of Care and the Non-perfectability of Man. *NEJM*, 295:1351–1355, 1976.

[6] McDonald CJ, Overhage JM. Guidelines You Can Follow and Can Trust. *JAMA*, 271:872–873, March 16, 1994.

[7] Shiffman RN. Towards Effective Implementation of a Pediatric Asthma Guideline: Integration of Decision Support and Clinical Workflow Support. In Ozbolt JG, ed. *Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care*, pages 797–801, 1994.

[8] Pryor TA. The HELP Medical Record System. *MD Computing*, 5:22–33, 1988.

[9] McDonald CJ, Tierney WM, Overhage JM, Martin DK,, Wilson GA. The Regenstrief Medical Record System: 20 years of Experience in Hospitals, Clinics, and Neighborhood Health Centers. *M.D. Computing*, 9(4):206–217, 1992.

[10] Pryor TA, Hripcsak G. Sharing MLM's: An Experiment between Columbia-Presbyterian and LDS Hospital. In Safran C, ed. *Proceedings of the Seventeeth Annual Symposium on Computer Applications in Medical Care*, pages 399–403, 1993.

[11] Agency for Health Care Policy and Research . *Urinary Incontinence in Adults*. U.S. Department of Health and Human Services, 1992.

[12] National Cholesterol Education Program . *Second Report of the Expert Panel on Detection, Evaluation, and Treatment of High Blood Cholesterol in Adults (Adult Treatment Panel II)*. National Institutes of Health, 1993.

[13] Lieberman H. Using Prototypical Objects to Implement Shared Behavior in Object Oriented Systems. *Sigplan Notices*, 21:214–223, 1986.

[14] Miller PL. *Expert Critiquing Systems: Practice-Based Medical Consultation by Computer*. Springer-Verlag, New York, NY, 1986.