MEDINFO 2004
M. Fieschi et al. (Eds)
Amsterdam: IOS Press

A Context-Adaptable Approach to Clinical Guidelines

Paolo Terenziani[a], Stefania Montani[a], Alessio Bottrighi[a], Mauro Torchio[b], Gianpaolo Molino[b], Gianluca Correndo[c]

[a]*Dipartimento di Informatica, Univ. Piemonte Orientale "A. Avogadro", Alessandria, Italy*
[b]*Laboratorio di Informatica Clinica, Az. Ospedaliera S. G. Battista, Torino, Italy*
[c]*Dipartimento di Informatica, Universita' di Torino, Torino, Italy*

Abstract

One of the most relevant obstacles to the use and dissemination of clinical guidelines is the gap between the generality of guidelines (as defined, e.g., by physicians' committees) and the peculiarities of the specific context of application. In particular, general guidelines do not take into account the fact that the tools needed for laboratory and instrumental investigations might be unavailable at a given hospital. Moreover, computer-based guideline managers must also be integrated with the Hospital Information System (HIS), and usually different DBMS are adopted by different hospitals. The GLARE (Guideline Acquisition, Representation and Execution) system addresses these issues by providing a facility for automatic resource-based adaptation of guidelines to the specific context of application, and by providing a modular architecture in which only limited and well-localised changes are needed to integrate the system with the HIS at hand.

Keywords:

Decision support systems, clinical, computer based guidelines, contextualization, three-layered architecture.

Introduction

Roughly speaking, clinical guidelines are a means for specifying the "best" clinical procedures and for standardizing them. Many different analyses have shown the advantages of adopting guidelines as a support for improving physician's work and/or optimizing hospital activities (see, e.g., [1, 2, 3]),. For instance, an in-depth analysis of critical factors for success of clinical guidelines has been proposed by Tierney et al. [4]. Morever many computer-based approaches of clinical guidelines have been built (consider, e.g., Asgaard [5], EON [6], GEM [7], GLIF [8], GUIDE [9], PROforma [10], and [1, 11]).

Usually, clinical guidelines are developed/revised by committees of expert physicians, and all the possible alternatives are deliberately described in order to make the guideline as general as possible. However, one of the biggest issues in guideline dissemination nowadays is the need of taking into account the presence of *local constraints* ([12]). The nature of such constraints can be related to local settings (resources availability, local best practices), to available practitioners' skills, or to patients' boundaries ([13]). This problem led, in some cases, to the design of very specific guidelines suitable for the specific context. However such guidelines were very far from being "standards" and were hardly sharable among different clinical structures. Thus, to preserve clinical guidelines generality and sharability, and, at the same time, to provide explicit support to their exploitation in specific context, the problem of *context adaptation* must be faced.

First of all, context adaptation involves taking into account the *resource*[1] requirements associated with each action (and, therefore, to each alternative path) in a guideline: only those actions (paths) whose resource requirements can be satisfied in the given context (e.g., hospital) can be executed. Thus, the context adaptation process could prune all non-executable actions (paths) from general guidelines.

Secondly, also the problem of adaptation to the *software* context has to be considered. Both in the acquisition and in the execution phase, computer-based managers of clinical guidelines should strictly interact with the Hospital Information System (HIS). For instance, the execution of a guideline on a patient must be based on the patient's data, which are stored into the patient Database. Thus, clinical guidelines managers should be adapted to operate at different hospitals, and, therefore, with possibly different DBMS (e.g., Access vs. Oracle vs. Caché).

In this paper, we discuss how we extended GLARE (Guideline Acquisition, Representation and Execution), a computer-based manager of clinical guidelines [14, 15, 16], in order to cope with context adaptation. As regards software adaptation, in section 3 we show how GLARE's three-layered modular architecture has been structured in such a way that only limited and local changes are necessary to allow the system interaction with different DBMS. As regards resource-based adaptation, in section 4 we describe an automatic pre-compilation tool which, taking into account a general clinical guideline (expressed according to GLARE's formalism) and a list of resources (e.g., the resources of a given hospital), gives as a result a context-based clinical guideline, in which all non-executable actions (paths) are automatically pruned.

1. When dealing with "resources", in this paper we basically mean the availability of tools needed for laboratory and instrumental investigations. Notice, however, that also time and/or cost constraints could be considered.

Finally, in section 5, we propose our conclusions, and discuss related work. We start with a brief presentation of GLARE, in the next section.

Main Features of GLARE

Representation formalism.

In order to guarantee usability of the program to physicians not expert in Computer Science, in GLARE we aimed at defining a *limited set* of clear representation primitives, covering most of the relevant aspects of a guideline [15]. We distinguish between *atomic* and *composite actions* (*plans*), where *atomic actions* represent simple steps in a guideline, and *plans* represent actions which can be defined in terms of their components via the *has-part* relation. The *has-part* relation supports top-down refinement: a guideline itself can be seen as a composite action. *Control relations* establish which actions can be executed next, and in what order. We distinguish between four different control relations: *sequence, controlled, alternative* and *repetition.*

Four different types of *atomic actions* have been defined as well: *work actions* (actions to be performed at a certain step of the guideline), *query actions* (requests for information), *decisions* (selections among alternatives) and *conclusions* (explicit output of a decision process). Actions are described in terms of their attributes.

Acquisition and Execution tools.

As in most approaches in the literature, GLARE distinguishes between the *acquisition* phase (when a guideline is introduced into the system –e.g., by a committee of expert physicians) and the *execution* phase (when a guideline is applied to a specific patient). Therefore, the system is composed by two main modules, the *acquisition tool* and the *execution tool.* The tools strictly interact with a set of databases (see section 3).

The acquisition tool provides a graphical interface to acquire atomic actions, *has-part* relations and control relations between the components of plans. The guideline is depicted as a graph, where each action is represented by a node (different forms and colours are used to distinguish among different types of actions), while control relations are represented by arcs. By clicking on the nodes in the graph, the user can trigger other windows to acquire the internal descriptions (attributes) of the nodes. The interface also shows the hierarchical structure of the guideline in the form of a tree, where plans can be seen as parents of their components (see figure 1).

The acquisition tool provides different forms of consistency checking, including *name* and *range checking,* and the check of several *logical design criteria* (for example, alternative arcs may only exit from a decision action).

The execution tool is typically used "on-line": a user physician applies a guideline with reference to a specific patient. This method is used for integrating guidelines into clinical practice [14]. Moreover, GLARE is available for "off-line" execution, i.e. for education, critical review and evaluation purposes [14]. The execution tool also incorporates a decision support facility, which allows physicians navigate through the guideline to see and compare alternative paths (stemming from decision actions) [16].

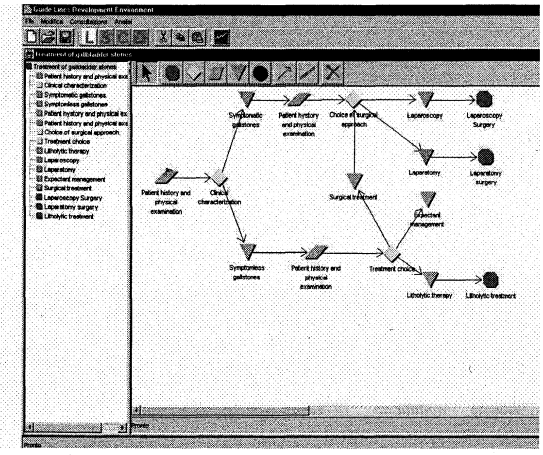

Figure 1 - A window of GLARE's acquisition tool graphical interface (concerning part of the gallbladder stones treatment guideline): on the left, the hierarchical structure of the guideline is displayed; on the right, the representation of control relations is shown in form of a graph

Testing.

We have already tested our representation formalism and acquisition tool prototype. Several groups of expert physicians, following a few-hour training session, used GLARE to acquire algorithms concerning different clinical domains (e.g., bladder cancer, reflux esophagitis, and heart failure), with the help of a knowledge engineer. In all the tests, our representation formalism and acquisition tool proved expressive enough to cover the clinical algorithms, and the acquisition of a clinical guideline was reasonably fast (e.g., the acquisition of the guideline on heart failure, starting from a non-structured textual representation, required only 3 days).

GLARE's Three-Layered Architecture

The overall GLARE's architecture is a three-layered one (see figure 2).

The highest layer (*system layer*) is composed by the two main modules described in section 2. In particular, the acquisition tool manages the representation of clinical guidelines, which are physically stored into a dedicated database, called CG DB. Moreover, it interacts with four additional databases: the Clinical DB, which provides a "standard" terminology to define the actions in the guideline; the Pharmacological DB, containing a list of drugs; the ICD DB, providing an international coding of diseases, and the Resource DB, that gives information about the resources available at the specific hospital where the guideline is meant to be used. On the other hand, the execution tool executes a guideline (previously acquired and stored in the CG DB) on a given patient, strictly interacting with the user-physician through its interface, and retrieving data from the Patient DB.

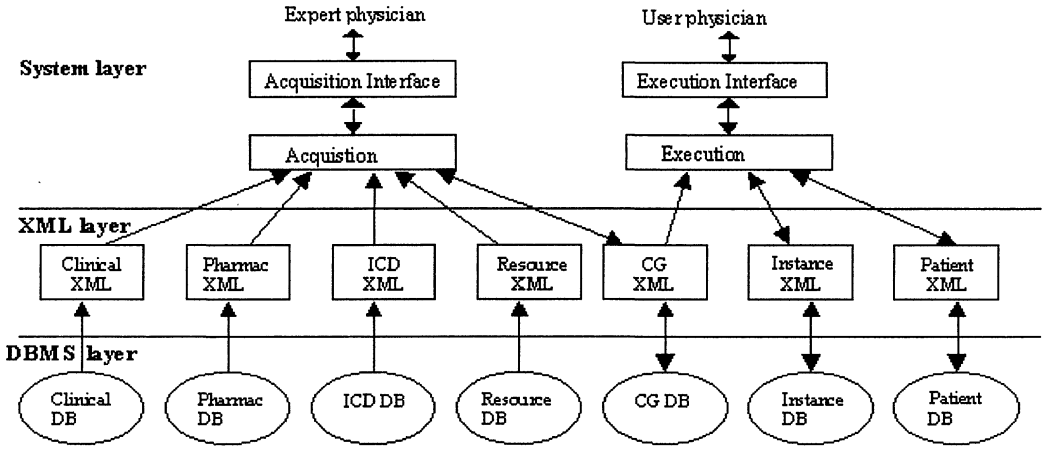

Figure 2 - GLARE's three-layered architecture

Different guidelines can be executed on the same patient (and, obviously, on different patients). The tool stores the state of each execution into the Instance DB.

The lowest layer of the architecture (*DBMS layer*) contains the DBMS, that physically stores the different databases described above, while the intermediate layer (*XML layer*) consists of a set of XML documents (one for each database). XML acts as an interlingua between the system layer and the DBMS layer: the acquisition and execution modules actually interact only with the XML layer, through which they obtain the knowledge stored into the DMBS. The use of XML as an interlingua allows us to express the guidelines in a format with characteristics of legibility, and to publish them on the web, rendering easy their dissemination. On the other hand, the DBMS layer grants a homogenous management of the data, by integrating the guideline representation with the pre-existent HIS in the same physical DBMS.

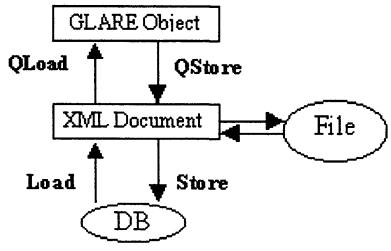

Figure 3 - The flow of data in GLARE's three-layered architecture

The XML layer and the DBMS layer manage the same knowledge, but they offer different functionalities. The XML layer manages small amounts of data and allows a quick interaction with the system (via the Qload – Quick Load - and QStore – Quick Store -functions; see figure 3). On the other hand, storing and loading data from the DBMS to the XML layer (Load and Store functions in figure 3) is a slower process, but it has to be performed only at the beginning of a working session, when the required guideline is retrieved, and at the end, when the guideline itself is saved[1]. Moreover, a DBMS is obviously able to manage large amounts of data, and provides a powerful query language. In such a way, we merge the advantages of exploiting an intermediate sharable data representation (through XML) and the DBMS technology.

However, the main reason for building such a ourour three-layered architecture is that it makes the system independent of the commercial DBMS adopted at the particular hospital. In fact, the interaction between the DBMS and the XML layer is devoted to a single software module (a Java package). Changing the DBMS only requires to modify such module and these changes are quite limited and well-localised. Thus, GLARE's three-layered architecture offers a useful support for adapting the tool to the software context.

Guideline's Resource-Based Adaptation

In this section, we present GLARE's facility to adapt guidelines on the basis of the resources available in a given context. In GLARE's representation formalism, the resources needed by each action of the guideline are explicitly declared. This means that a check can be performed in order to prune the branches of the guideline that cannot be executed because of resources unavailability.

In fact, a clinical guideline describes a set of alternative paths that the physicians can choose during the diagnostic/therapeutic process. Pruning non-executable alternative branches brings out with a context-dependent guideline, that describes all and only those actions that respect the original meaning of the general in-

1. Notice that the "XML Document" box in figure 3 stands for the XML document representing the guideline in main memory, while "File" denotes its representation (as an XML file) in secondary memory.

put guideline, and that can actually be implemented in the given context (since the required resources are available).

Informally, a *legal path* in a guideline is a path from the starting action to an ending action such that all the resources needed by (the actions in) the path are available. We base our procedure on an inductive definition: an action belongs to a *legal path* if :

- it does not require unavailable resources and
- it is the last node of a path or, alternatively, there is at least one action that follows it which belongs to a legal path.

In GLARE's formalism, the only actions that require resources are *work* actions and *query* actions, and the only actions from which alternative paths may stem are *decision* actions. However, for the sake of simplicity we will describe the procedure assuming that all the actions may require a (possibly empty) set of resources and that all the actions are followed by a (possibly empty) set of actions.

Our approach is to rebuild a general guideline from the starting action collecting only those paths that require available resources. The procedure takes as input a guideline **GL** and a set of available resources and produces as output an adapted guideline $\mathbf{GL_{context}}$ that contains only the legal paths of **GL**. In order to do this, all the actions in **GL** are taken into account.

Given the current action A in **GL**, the algorithm operates as follows:

1. if A is already present in $\mathbf{GL_{context}}$ (i.e., A belongs also to another legal path in $\mathbf{GL_{context}}$), no other actions are required;
2. if not, our procedure checks the set of resources required by A. If A exceeds the available resources the procedure returns a failure. Otherwise, A could belong to a legal path;
3. if A is a final action, it is added to $\mathbf{GL_{context}}$;
4. if A is not a final one, an additional recursive check is applied to every action which follows A in **GL**, so that all the legal actions following it are collected. In the case that there are no legal actions following A, the procedure fails to find a legal path including A;
5. once collected all the legal actions following A, the type of A is taken into account;
6. if A is a composite action, the procedure is recursively applied to its components. If such guideline has at least one legal path:
   - a copy of A is done (with the adapted internal guideline within);
   - the set of actions following A is updated with the set of legal actions that follow A previously collected by the recursive calls;
   - the copy of A is added to $\mathbf{GL_{context}}$ and returned as a result.

   Otherwise the procedure returns a failure;
7. if the action A is of any other type, no other control is performed. A copy of A is done, and the set of legal actions that follow A is updated and, finally, the copy of A is added to $\mathbf{GL_{context}}$ and returned as a result.

Comparisons, Conclusions and Future Work

Guidelines dissemination and integration into clinical practice should recognize the multiplicity of working settings and information systems environments within which the guidelines themselves are meant to be implemented [17]. As a consequence, some of the approaches in the literature started considering the problem of adapting clinical guidelines to the context of application.

As concerns the adaptation based on resources availability, one abstract solution is to have a high level description of the guideline's intentions, in order to ensure the adaptability of the procedure to different contexts still preserving the guideline's intentional objectives (see [18]). Such an approach has been followed in CAMINO (see [19]), a tool that provides users with a user-friendly interface to modify (e.g., by adding/removing/changing actions) a guideline, using additional information about the hospital.

[13] proposes an approach in which the dependencies between actions in a guideline can be explicitly described, and where users' modifications of a general guideline must respect such dependencies.

The above solutions provide facilities to help physicians modify guidelines (consistently with guidelines' intentions and/or functional dependencies). On the other hand, GLARE's resource-based adaptation tool is a completely automatic one: it takes as input a general guideline and the list of available resources, and automatically prunes out non-executable alternative paths. Moreover, in GLARE, we also take into account the problem of adaptation to the software environment: GLARE's three-layered architecture makes easier the task of integrating our system with different commercial DBMS, thus providing a useful support to software contextualization.

References

[1] C. Gordon and J.P. Christensen, Health Telematics for Clinical Guidelines and Protocols, IOS Press, 1995.

[2] G. Molino, From Clinical Guidelines to Decision Support, Proc. Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, LNAI 1620, Aalborg, 3-12, 1999.

[3] I. Purves, Computerised Guidelines in Primary Health Care: Reflections and Implications, in: C. Gordon and J.P. Christensen, eds., Health Telematics for Clinical Guidelines and Protocols, IOS Press, Amsterdam, 57-74, 1995.

[4] Tierney WM, Overhage JM, McDonald CJ. Proceedings of the 20th Annual AMIA Fall Symposium 1996; 459-462.

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

[6] M.A. Musen, S.W. Tu, A.K. Das, Y. Shahar, EON: a component-based approach to automation of protocol-directed therapy, JAMIA, 3(6), 367-388, 1996.

[7] R.N. Shiffman, B.T. Karras, A. Agrawal, R. Chen, L. Menco, and S. Nath, GEM: a proposal for a more comprehensive guideline document model using XML, JAMIA, 7(5), 488-498, 2000.

[8] M. Peleg, A.A. Boxawala, et al., GLIF3: The evolution of a Guideline Representation Format, Proc. AMIA 2000, 645-649.

[9] S. Quaglini, M. Stefanelli, A. Cavallini, G. Miceli, C. Fassino, and C. Mossa, Guideline-based careflow systems, Artificial Intelligence in Medicine, 20(1), 5-22, 2000.

[10]J. Fox, N. Johns, A. Rahmanzadeh, R. Thomson, Disseminating medical knowledge: the PROforma approach, Artificial Intelligence in Medicine, 14, 157-181, 1998.

[11]Special Issue on Workflow Management and Clinical Guidelines, D.B. Fridsma (Guest ed.), JAMIA, 22(1), 1-80, 2001.

[12]M.D. Cabana, C.S. Rand, N.R. Powe, A.W. Wu, M.H. Wilson, P.C. Abboud, H.R. Rubin, Why don't physicians follow clinical practice guidelines? A framework for improvement. JAMA, 282(15), 1458-1465, 1999.

[13]A.A. Boxwala, Applying axiomatic design methodology to create guidelines that are locally adaptable, Proc. AMIA 2002.

[14]P. Terenziani, F. Mastromonaco, G. Molino, M. Torchio, Executing clinical guidelines: temporal issues, Proc. AMIA 2000, 848-852.

[15]P. Terenziani, G. Molino, and M. Torchio, A Modular Approach for Representing and Executing Clinical Guidelines, Artificial Intelligence in Medicine, 23, 249-276, 2001.

[16]P. Terenziani, S. Montani, A. Bottrighi, G. Molino, M. Torchio. Supporting physicians in taking decisions in Clinical Guidelines: the GLARE's "what if" facility. Proc. AMIA 2002, 772-776.

[17]A.A. Boxwala, S. Tu, M. Peleg, Q. Zeng,O. Ogunyemi , R.A. Greenes, et al., Toward a representation format for sharable clinical guidelines, Journal of Biomedical Informatics, 34(3), 157-169, 2001.

[18]Y. Shahar, S. Miksch, P. Johnson, An intention-based language for representing clinical guidelines, Proc AMIA 1996, 592-6.

[19]D.B. Fridsma ,J.H. Gennari, M.A. Musen, Making generic guidelines site-specific, Proc. AMIA 1996, 597-601.

Address for correspondence

Terenziani Paolo,

Dipartimento di Informatica, Univ. Piemonte Orientale

"A. Avogadro", Spalto Marengo 33, 15100 Alessandria, Italy;

phone +39 0131 287447 fax: +39 0131 287440