Contextual and Temporal Clinical Guidelines

Augusto Guarnero*, Marina Marzuoli*, Gianpaolo Molino*,
Paolo Terenziani§, Mauro Torchio*, Katia Vanni§

§Dipartimento di Informatica,
Universita' di Torino
Corso Svizzera 185, 10149 Torino, Italy

*Laboratorio di Informatica Clinica
Az. Ospedaliera S. Giovanni Battista
Corso Bramante 88, 10126 Torino, Italy

ABSTRACT

In this paper, we propose an approach for managing clinical guidelines. We sketch a modular architecture, allowing us to separate conceptually distinct aspects in the management and use of clinical guidelines. In particular, we describe the clinical guidelines knowledge representation module and we sketch the acquisition module. The main focus of the paper is the definition of an expressive formalism for representing clinical guidelines, which allows one to deal with the context dependent character of clinical guidelines and takes into account different temporal aspects.

1 INTRODUCTION

The dissemination of clinical guidelines forms one policy for targeting areas of healthcare where there is clear evidence of what constitutes best practice, and may provide an effective means to implement workable policies for improving clinical health care. The advantages of the introduction of guidelines in the clinical framework is discussed, e.g., in [7,8,11]. Many different systems and projects have been developed to deal with clinical guidelines (consider, e.g., [9], Asgaard[12], DILEMMA [10], OPADE [4], PROforma [5]). In this paper, we describe Clinical Guidelines Manager, a system for building and representing clinical guidelines, and for using them in different tasks. Section 2 sketches the architecture of the system. Section 3, which is the core of the paper, focuses on the representation actions in guidelines, proposing a general and task-independent ontology to deal with clinical actions. Finally, Section 4 sketches the acquisition module, and Section 5 proposes conclusions and discussion.

2 A MODULAR APPROACH

The overall problem of automatically managing clinical guidelines is a very complex one. In fact, clinical guidelines may play very different roles in the clinical process. For example, they can be used as a system to support physicians in the treatment of diseases, or as a system of critic or evaluation. Thus, the definition of an incremental and modular strategy to build an automatic manager of clinical guidelines is advantageous from both the practical and the conceptual point of view. In fact, it allows system designers to focus on well-defined subparts of the overall problem. The core of the modular approach we are building is CG_KRM (Clinical Guidelines Knowledge Representation Manager), which manages the internal representation of clinical guidelines, and operates as a knowledge server for the other modules. The assumption of our approach is that knowledge in the clinical guidelines is independent of the use (e.g., support, evaluation etc.) Thus, CG_KRM operates as a task-independent knowledge server for the other modules, and each one of these modules is devoted to a specific task. The Clinical Guidelines Acquisition Manager (CG_AM) provides physicians with a user-friendly graphical interface in order to introduce and describe the clinical guidelines into the CG_KRM. The Clinical Guidelines Support System (CG_SS in Figure 1) applies guidelines for supporting physicians; the Clinical Guidelines Evaluation System (CG_ES in Figure 1) detects the differences between suggested guidelines and the procedure actually adopted by physicians. These modules strictly interact with the CG_KRM, taking from it all the knowledge about guidelines stored by physicians. The overall architecture of the system is shown in Figure 1.

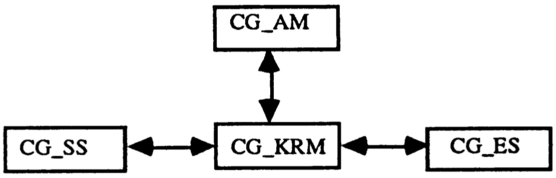

Figure 1. Architecture of the system

Notice that the architecture in Figure 1 is an open (and modular) one, since we intend to add new modules and functionalities to the system. CG_SS and CG_ES are still under development, and will not be discussed in the following.

3 REPRESENTING CLINICAL GUIDELINES

In the following, we propose the representation formalism we devised for medical guidelines.

1091—8280/98/$5.00 © 1998 AMIA, Inc.

3.1 Ideal vs contextual guidelines

In most cases, the contextual character of (clinical) guidelines has not been taken seriously into account. In fact, most clinical guidelines are "ideal", in the sense that they describe one or more alternative procedures which constitute the "nearly optimal" way of operating (e.g., of treating a given disease or of proceeding for finding a diagnosis, given a typical set of observations and findings). However, often "ideal" guidelines can not be executed by physicians in actual hospitals, since different sources of contextual limitations need to be faced. A typical source of contextual limitation is the availability of resources (e.g., CT), which may be not present or not available in the specific hospital. Moreover, budget and/or time constraints may make some "ideal" clinical procedure unfeasible or, at least, not recommended. Thus, we believe that explicitly representing contextual parameters such as resource availability, times and costs is a crucial step towards the actual applicability of clinical guidelines.

3.2 Atomic vs composite actions

In our work, we focus our attention on the notion of clinical action, which is a basic notion for describing clinical guidelines. We use the term "action" in a quite broad sense in order to indicate the different activities which may characterise a diagnostic task, or the application of a given therapy, or the finding/retrieving of the values for a given set of findings, or other clinical activities. In this broad sense, a guideline itself can be conceived as a complex action, composed by a given number of elementary actions. We distinguish between atomic and composite actions. Atomic actions can be regarded as elementary steps in a guideline, in the sense that they do not need a further de-composition into sub-actions in order to be executed. Composite actions are composed by other actions (elementary or not). Notice that the atomicity of an action is not an absolute feature: the physician introducing the guidelines can choose the level of detail of the description and, thus, can choose which actions are atomic and which are not.

3.3 Atomic actions

The internal structure of the description of an atomic action is shown in Figure 2. Only the name of the action is compulsory: all the other parts of the description may be omitted by the physician.

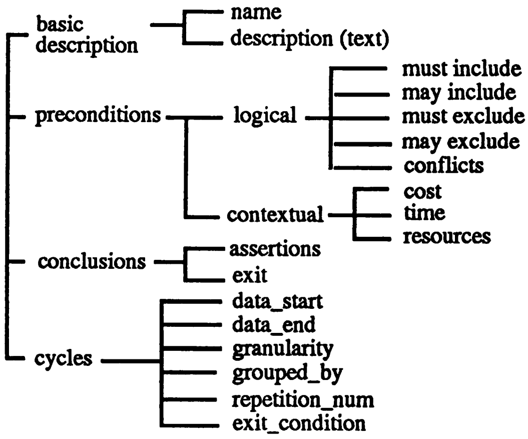

Figure 2. Atomic actions

The **basic description** of an action consists of its **name** and of an additional **description** of the action itself, which is a string of text. Actions can be described on the basis of three different types of properties: pre-conditions, conclusions and cycles. **Pre-conditions** state the conditions under which a given action can be applied or not. In particular, contextual pre-conditions allow the physician to specify minimum and maximum **cost** of the given action (see also the OPADE project [4]); minimum and maximum **time** required to perform the given action; the **resources** (e.g., specialised instruments for laboratory tests) needed in order to perform the action. These attributes may be used for establishing the actual applicability of an action in a given context (e.g., resource and/or cost limitations), and are also very important when guidelines are used in order to check the quality (in terms of consts and times) of the hospitalisation service.

Logical pre-conditions consist of conflicts, and must/may include/exclude preconditions. **Conflicts** state the list of incompatible and conflicting actions (not to be performed together with the one being described). **Including and excluding pre-conditions** are logical formulae stating the conditions which must hold (not hold) for the action to be executable. We distinguish between **must** and **may** preconditions: for example, a "must include" precondition is a condition which must hold for the action to be executed, while the action can also be performed in case some of its "may include" preconditions do not hold (in such a case, we have some belief against the application of the action). The case of excluding pre-conditions is analogous.

Conclusions are the effects of the execution of the action. If the action is successfully performed, than the new facts specified into **assertions** hold, and can be inserted into the clinical database. Besides facts, assertions may contain *functions* for evaluating, e.g., the degree of beliefs some of the facts in the assertions, or even rules of the general form *IF condition THEN assertions*. These rules allow one to conclude the facts in the *assertions* part of the rule only in case *condition* part of the rule hold. On the other hand, the **exit** property specifies alternative actions to be executed in case the given action fails. Such alternative actions can be specified in the *THEN* part of rules of the general form *IF condition THEN action*. In such a case, conditions in the *IF* part of the rules are used to test for which reason the described action failed, or to manage exceptional conditions (e.g., vital signs monitoring).

Finally, an important feature of clinical guidelines is the occurrence of repeating actions. In general, there are two different ways to specify the repetitions to be executed. One way is to state an **exit condition**, i.e., to say that the action has to be performed until the given condition remains true. The other way is to specify a frame of time for the repetitions, starting from a **start date** and ending to an **ending date**. In both cases, however, the physician often needs to specify the frequency of the repetitions in time. In general, specifying frequencies could require a quite complex formalism. Consider, e.g., the frequency "3 times every 2 days". One has to specify both the **granularity** chosen for the repetition (days in the example) and the **groupings** to be considered for the granularity (3 in the example) in order to define the periodicity of the action. Moreover, the number of repetitions of the action (**repetition_num**) in the given periodicity must also be indicated (2 in the example).

3.4 Composite actions

Composite actions are actions which, in turn, are composed by other actions, which may be elementary or not. In general, in our approach, even a whole guideline can be represented as a composite action. In such a way, we enforce a hierarchical structure upon guidelines, leading to a top-down incremental refinement of the guideline by the physician describing it.

The description of a composite action consists of two parts. The first part concerns the general description of the action itself, and is analogous to the description of atomic actions (see Figure 2).

The second part is the **structural description**, which is the description of the way in which the subactions are composed in order to form the composite actions itself. We pointed out three main classes of structures for composite actions: sequences, concurrent actions and alternatives.

A **composed sequence action** is a set of n≥2 sub-actions $A_1, \dots, A_n$ to be executed in sequence. This means that, for each i, 1≤i≤n, the execution of A_{i+1} must start after the end of the execution of A_i. We also allow the possibility of stating the minimum and maximum **delay** between pairs of actions which are in a sequence (as in Asbru [12]).

If delays are specified, a temporal reasoning mechanism has to be introduced in order to check the consistency of the temporal constraints. In fact, in general, sequences of actions sharing some subactions may form graph structures, and the constraints on the minimum and maximum durations of actions and minimum and maximum delays between actions have to be propagated throughout the graph, to check consistency and to infer new constraints. In our approach, temporal reasoning is performed by LaTeR [1], a general-purpose temporal manager which performs correct and complete propagation of temporal constraints. LaTeR has been extended in order to deal directly with temporal constraints on tuple in relational databases, as described in [2], [3].

In some case, strict sequencing of actions is not needed. **Concurrent composite actions** are actions composed by n≥2 sub-actions, which can be performed concurrently. In particular, we intend that a composite action A composed by n concurrent sub-actions $A_1, \dots, A_n$ can start with the concurrent execution of all its sub-actions, and terminates only when all of its sub-actions terminate. A concurrent action fails if at least one of its sub-actions fails.

Finally, in most cases, alternative strategies can be followed in order to achieve a given clinical task. **Composite alternative actions** are introduced in order to represents alternatives. An action A composed by the alternative sub-actions $A_1, \dots, A_n$ represents the fact that one among $A_1, \dots, A_n$ has to be executed. Thus, an alternative action fails only when all of its composing alternatives fail. Notice that, in general, each one of the alternative sub-actions has its pre-conditions, so that only those sub-actions whose preconditions hold can be executed. Moreover, in case of composed alternative actions, we also allow the physician the possibility of introducing **pro** and **cons** for each one of the alternative sub-actions, in order to improve the conflict resolution process whenever more than one alternative can be executed.

3.5 A practical example

As a simple example, let us consider the case when a reliever medication is requested for dispnea in a patient affected by bronchial asthma.

The following *sequence of actions* should be considered:

(A1) problem identification;

(A2) choice of the therapeutic strategy for each identified problem;

(A3) selection of a drug class within those reflecting the chosen strategy.

A1, A2 and A3 are composite actions. For example, A3 is a *composite action* resulting from 4 *atomic alternative actions*:

(A3.1) inhaled short-acting β2-agonist

(A3.2) oral short-acting β2-agonist

(A3.3) short-acting theophylline

(A3.4) inhaled anticholinergics

In the following we provide the description of A3.3 according to the schema in Figure 2. The description of A3.1, A3.2 and A3.4 is similar.

name: eligibility of the short acting theophylline treatment
description: evaluation of the compatibility of the chosen treatment with the clinical scenario
preconditions-logical:
must exclude: preexisting treatment with long acting theophylline
preconditions - contextual:
resources: drug availability
conclusions:
assertions: suitable/unsuitable

3.6 Implementation of CG_KRM

We started from the conceptual analysis before and we proposed an internal representation of actions and their descriptions using a relational database management system, namely ACCESS. The choice of a relational database instead of, e.g., a representation formalism based on frames or on semantic networks (similar, e.g., to the one in DILEMMA [10]) was mainly due to the needs of exploiting efficient and engineered primitives for storing and retrieving data, and of providing a uniform representation for clinical guidelines and for the clinical and patient databases, which were already implemented as ACCESS relations. The importance of an homogeneous and integrated treatment of clinical records and clinical guidelines both at the physical and at the conceptual level has been stressed, e.g., in [6].

In the most general case where both minimum and maximum durations of actions and delays between sequences of actions are specified, CG_KRM also uses LaTeR [1,3], which operates on the temporal data stored in ACCESS in order to perform temporal reasoning and to check their consistency. However, LaTeR is only conceived as a service provided by CG_KRM, and is not visible for users.

4 KNOWLEDGE ACQUISITION MODULE

The Clinical Guidelines Acquisition Manager (CG_AM) provides a graphical interface for the acquisition process, allowing physicians to introduce and modify guidelines in a more natural way.

The interaction starts with the request of inserting a new guideline, which is treated as a composite action. The description of an atomic action and the description part of a composite action can be easily introduced by the physician: in such a case, CG_AM provides a sequence of popup windows asking for all the properties shown in Figure 1, and the physician has simply to fill the desired slots. On the other hand, CG_AM provides a set of graphical primitives to introduce the structural description of composite actions. Given a composite action, each one of its composing sub-action can be drawn as a node of a graph. Arcs in the graphs represent the structure: different arcs are provided in order to represent the fact that the connected nodes are in sequence, concurrent or in alternative.

CG_AM also provides physicians with a special window showing the tree representing the hierarchical structure of the overall guideline being built. The tree can be used to browse the guideline, as well as to build it. In fact, selecting a node in the tree automatically pops up a window which shows the structural description of the action (if already given by the physician) or which allow the physician to introduce it. Moreover, whenever a new node is introduced by the physician in the graph representing the structural description of an action, the corresponding node is automatically added by CG_AM in the tree. For example, Figure 3 reports the tree for the part of clinical guideline described in Section 3.5. In Figure 3, annotations "Seq" and "Alt" stands for Sequence and Alternative resprectively, and names of actions have been abbreviated.

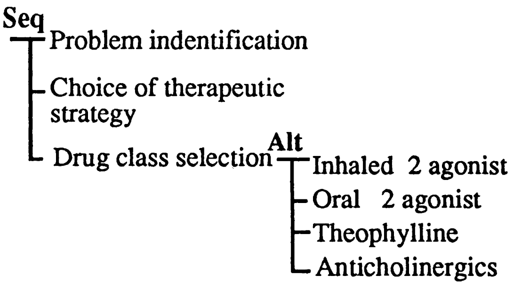

Figure 3. Tree for the guideline in Section 3.5

CG_AM also supports primitives for modifying, cutting, pasting and deleting (part of) the description of actions, and of structural descriptions.

5 CONCLUSIONS AND DISCUSSION

In this paper, we propose an approach for managing medical guidelines. We first sketch a modular architecture, allowing us to separate conceptually distinct aspects in the management and use of clinical guidelines. We then focus on a part of the overall architecture, namely the knowledge server for representing clinical guidelines and the acquisition system to acquire them.

Different approaches to clinical guidelines have been developed within the Artificial Intelligence in Medicine literature (see the introduction). The distinction between modules dealing with knowledge representation and acquisition and inferential modules using guidelines for various tasks is also used, e.g., in DILEMMA [10] and PROforma [5]. In the Asgaard project [12], this leads to the distinction between design time (the time when guidelines are built, via an acquisition module) and execution time (the time when guidelines are used for different tasks, such as, e.g., support for the physician). We also maintain such a distinction, and in this paper we focus only with representations and design time issues.

PROforma [5] seems to us the approach closest to ours. In particular, the acquisition techniques are very similar. Some peculiar features of our approach, distinguishing it from PROforma are:

- the management of contextual parameters, to deal with the contextual character of clinical guidelines (the treatment of costs is also considered, e.g., in OPADE [4]).
- the management of temporal constraints between actions. To the best of our knowledge, an explicit and extensive treatment of temporal constraints on the durations and delays between actions is only proposed in Asbru [12], which also considers the distinction between concurrent actions and actions in sequence, as well as cyclic actions.

A peculiar feature of our approach is the lack of an explicit primitive in the representation in order to deal with conditions. The rationale underlying such a choice is the following: we believe that, in clinical guidelines, conditions do not play an important role per se, but only in conjunction with the corresponding actions. Thus, in our approach, we distinguish among different types of conditions (see section 3.3), but we associate conditions directly to the corresponding actions. Besides improving the homogeneity of our approach, this choice allows us to represent in a quite natural way a very frequent structure in medical guidelines: the choice of one among several alternatives on the basis of some precondition. Consider, e.g., the case when one wants to execute action A1 if its pre-condition C1 holds, or A2 if C2 holds, ... or An if Cn holds. In our formalism, this can be easily represented as an alternative composite action A = A1(C1), A2(C2), ..., An(Cn), where Ai(Ci) represents the action Ai with its associated pre-conditions Ci. Consider, for example, the definition of the composite alternative action "drug class selection" in the example in Section 3.5: A3.1 or ... or A3.4 are selected, depending on the satisfiability of their preconditions (e.g., A3.3 cannot be selected in case of preexisting treatment with long acting theophylline treatment). On the other hand, in most approaches (consider, e.g., PROforma [5]) one should be forced to introduce an undesired ordering in the alternatives, and to express them in some sort of cascade of IF THEN ELSE structures (e.g., IF C1 THEN A1 ELSE IF C2 THEN A2 ELSE .... ELSE IF Cn THEN An).

Currently, CGM is used in order to deal with clinical guidelines about stroke.

REFERENCES

[1] V. Brusoni, L. Console, B. Pernici, and P. Terenziani. Later: an efficient, general purpose manager of temporal information. IEEE Expert, 56-64, July/August 1997.

[2] V. Brusoni, L. Console, F. Molino, G. Molino, E. Nicolosi, P. Terenziani. Clinical-Mate: a manager of temporal databases for clinical applications. JAMIA, Proc. AMIA Fall Symposium, pag. 893, 1997.

[3] V. Brusoni, L. Console, B. Pernici, P. Terenziani. Qualitative and Quantitative Temporal Constraints and Relational Databases: Theory, Architecture, and Applications. Accepted for pubblication in IEEE Transactions on Knowledge and Data Engeneering.

[4] I. de Zehger, C. Milstein, B. Sene, A. Venot. Prescription Guidelines in OPADE: what are they, how are they used? in [9], 199-205.

[5] J. Fox, N. Johns, A. Rahmanzadeh, R. Thomson. PROforma: a method and language specifying clinical guidelines and protocols. Proc. MIE'96, pag. 516-520, 1996.

[6] A. Glowinski. Integrating guidelines and the clinical record: the role of semantically constrained terminologies. in [9], 207-218.

[7] C. Gordon. Practice Guidelines and healthcare telematics: towards an alliance. In [9], 3-15.

[8] J.M. Grimshaw, I.T. Russel. Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluation. Lancet Vol 342342, 1317-1322, 1993.

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

[10] S.I. Herbert. Informatics for Care Protocols and Guidelines: Towards a European Knowledge Model. in [9], 27-42.

[11] I. Purves. Computerised Guidelines in Primary Health Care: Reflections and Implications. in [9], 57-74.

[12] Y. Shahar, S. Miksch, P. Johnson. A Task-Specific Ontology for Design and Execution of Time-Oriented Skeletal Plans. Report SMI-96-0633, Stanford Univ. School, Stanford, 1997.