Applying Artificial Intelligence to Clinical Guidelines: the GLARE Approach *

Paolo Terenziani[1], Stefania Montani[1], Alessio Bottrighi[1], Mauro Torchio[2], Gianpaolo Molino[2], Luca Anselma[3], and Gianluca Correndo[3]

[1] DI, Univ. Piemonte Orientale "A. Avogadro"
Spalto Marengo 33, Alessandria, Italy
[2] Lab. Informatica Clinica, Az. Ospedaliera S. G. Battista
C.so Bramante 88, Torino, Italy
[3] Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, Torino, Italy

Abstract. In this paper, we present GLARE, a domain-independent system for acquiring, representing and executing clinical guidelines. GLARE is characterized by the adoption of Artificial Intelligence (AI) techniques at different levels in the definition and implementation of the system.
First of all, a high-level and user-friendly knowledge representation language has been designed, providing a set of representation primitives.
Second, a user-friendly acquisition tool has been designed and implemented, on the basis of the knowledge representation formalism. The acquisition tool provides various forms of help for the expert physicians, including different levels of syntactic and semantic tests in order to check the "well-formedness" of the guidelines being acquired.
Third, a tool for executing guidelines on a specific patient has been made available. The execution module provides a hypothetical reasoning facility, to support physicians in the comparison of alternative diagnostic and/or therapeutic strategies. Moreover, advanced and extended AI techniques for temporal reasoning and temporal consistency checking are used both in the acquisition and in the execution phase.
The GLARE approach has been successfully tested on clinical guidelines in different domains, including bladder cancer, reflux esophagitis, and heart failure.

1 Introduction

Clinical guidelines represent the current understanding of the best clinical practice, and are now one of the most central areas of research in Artificial Intelligence (AI) in medicine and in medical decision making (see, e.g. [5, 7, 8, 12]). Clinical guidelines play different roles in the clinical process: for example, they can be used to support physicians in the treatment of diseases, or for critiquing, for evaluation, and for education purposes. Many different systems and projects have been developed in recent years in order to realize computer-assisted management of clinical guidelines (see e.g.,

* This is a preprint of a paper published in Lecture Notes in Artificial Intelligence, 2829:536-547, Springer-Verlag, ©2003 http://www.springer.de/comp/lncs/.

Asbru [15], EON [10], GEM [16], GLARE [19–21], GLIF [11], GUIDE [14], ONCOCIN [22], PROforma [4], T-HELPER [9], and also [6, 2]).

The overall challenge of designing and implementing such tools is very complex. In this paper we show how in the GLARE system the adoption of AI techniques provides relevant advantages, especially from the point of view of the user-friendliness of the approach (a more detailed description of GLARE's basic features can be found in [20]). GLARE's architecture is sketched in section 2. In section 3, we highlight GLARE's representation formalism. Section 4 and section 5 describe the acquisition tool and the execution tool functionalities respectively, with specific attention to the treatment of temporal constraints. Section 6 sketches some testing results. Finally, section 7 presents comparisons and conclusions.

2 Architecture of GLARE

The overall GLARE's architecture is a three-layered one (see figure 1).

The highest layer (*system layer*) is composed by two main modules, the *acquisition tool* and the *execution tool*. Both tools need to access data stored into a set of databases. 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: the Pharmacological DB, storing a structured list of drugs and their costs; the Resource DB, listing the resources that are available in a given hospital (it is therefore used to represent the context-dependent version of a guideline); the ICD DB, containing an international coding system of diseases; the Clinical DB, providing a "standard" terminology to be used when building a new guideline, and storing the descriptions and the set of possible values of clinical findings. The interaction with the Clinical DB during acquisition allows for standardization (since experts are forced to use the same vocabulary) and for correctness (since only values for findings that are compatible with the range of values fixed in the Clinical DB itself can be specified).

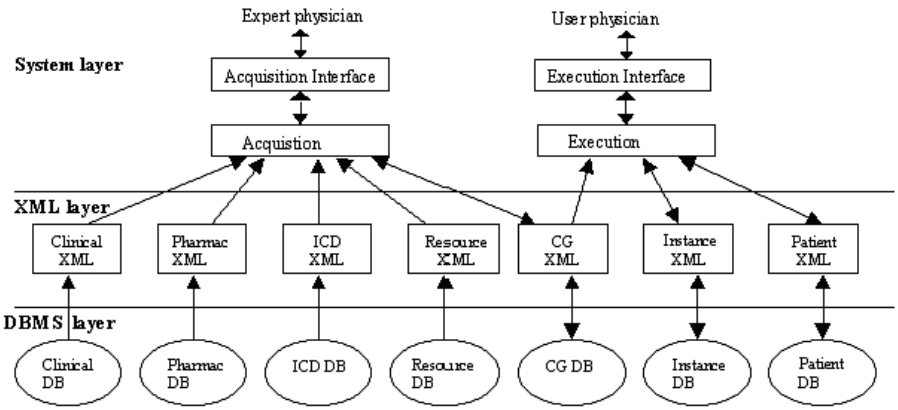

Fig. 1. GLARE's three-layered architecture

The execution module executes a guideline for a specific patient, taking into account the patient's data (automatically retrieved from a database called Patient DB). This tool stores the status of the execution in the Instance DB and interacts with the user-physician via a user-friendly graphical interface.

The lowest layer of the architecture (*DBMS layer*) is made by the DBMS, that physically stores the different databases described above. However, in GLARE, the interaction between the acquisition and the execution tools with such databases is not a direct one, since it is mediated by the introduction of an intermediate layer (*XML layer*). The 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 DBMS. 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 Hospital Information System in the same physical DBMS.

The three-layered architecture makes GLARE independent of the commercial DBMS adopted by 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-localized.

3 Representation Formalism

In order to guarantee usability of GLARE to user-physicians not expert in Computer Science, we have defined a *limited set* of clear representation primitives, covering most of the relevant aspects of a guideline. In particular, we have focused the attention on the concept of action, a basic primitive notion for describing clinical guidelines. We use the notion of "action" in quite a *broad sense*, in order to indicate the different activities which may characterize a diagnosis, or the application of a given therapy, or the finding/retrieving of information, or other clinical tasks. Given this notion, a guideline itself can be conceived as a complex action, composed by a 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 to be executed. *Composite actions* are composed by other actions (atomic or composite).

Four different types of *atomic actions* can be distinguished: *work actions*, *query actions*, *decisions* and *conclusions*. *Work actions* are atomic actions which must be executed at a given point of the guideline, and can be described in terms of a set of attributes, such as name, (textual) description, cost, time, resources, goals. *Query actions* are requests of information, that can be obtained from the outside world (physicians, databases, knowledge bases). *Decision actions* are specific types of actions embodying the criteria which can be used to select from alternative paths in a guideline. In particular, *diagnostic decisions* are represented as an open set of triples $\langle diagnosis, parameter, score\rangle$ (where, in turn, a $parameter$ is a triple $\langle data, attribute, value\rangle$),

plus a threshold to be compared with the different diagnoses' scores. On the other hand, *therapeutic decisions* are based on a pre-defined set of parameters: effectiveness, cost, side-effects, compliance, duration. Finally, *conclusions* represent the explicit output of a decision process (for instance, assuming a given diagnostic hypothesis is a typical conclusion of a diagnostic decision action).

Composite actions are defined in terms of their components, via the has-part relation (this supports for top-down refinement in the description of guidelines). On the other hand, a set of *control relations* establishes which actions might be executed next and in what order. We distinguish among four different control relations: *sequence*, *controlled*[1], *alternative* and *repetition*.

A distinguishing feature of GLARE is its capability of representing (and treating) temporal constraints. Temporal constraints play a fundamental role in both the description and the execution of clinical guidelines. We have worked to design a temporal representation formalism as expressive as possible, still maintaining the tractability of the temporal reasoning process. Our formalism allows one to represent the (minimum and maximum) duration of each non-composite action. Temporal constraints can also be associated with control relations between actions. In the sequence and alternative relations, one can indicate the minimum and/or maximum delay between actions. In a controlled relation, one can specify the minimum and/or maximum distance between any pair of endpoints of the actions involved. On the basis of such distances, one can express both *qualitative* constraints between actions (however, only *continuous pointizable relations* can be coped with [23]) and *quantitative* ones. Finally, two different ways of specifying repetitions are defined (and can be combined): one can state that the action has to be performed until a given *exit condition* becomes true, or can specify duration (*frame time*) for the repetitions. In both cases, the *frequency* of the repetitions in time has to be specified as well; then, several other parameters must/can be provided.

Ex.1 *For six months, perform action A twice each five days for twenty days, and then suspend for ten days.*

The *frame time* (henceforth called FT for short) can be defined as "*the interval which contains all the instances of the event*" [3], ("*for six months*" in Ex.1). The description of repeated periodic events splits FT into a sequence of intervals when actions are performed (called *action-times* - AT; "*twenty days*" in Ex.1) and "pause" intervals (*delay time* - DT; "*ten days*" in Ex.1). In turn, ATs are split into *I-times* (IT; "*five days*" in Ex.1) where actions are actually performed (if DT is null, AT coincides with IT). Finally, we call the number of actions in each I-time "*frequency*" ($freq$; two in Ex.1).

Besides these "explicit" constraints, also the implicit constraints implied by the has-part relations between actions have to be taken into account [18].

4 Acquisition

The acquisition module is a user-friendly tool that provides expert physicians with:

[1] Controlled relations are used to represent temporally constrained actions, such as "A during B", "start of A at least 1 hour after the beginning of B", and so on.

(i) a graphical interface, which supports primitives for drawing the control information within the guideline, and ad hoc windows to acquire the internal properties of the objects;
(ii) facilities for browsing the guideline;
(iii) "intelligent" help and consistency checking (see next subsection).

4.1 Consistency Checking

The acquisition tool provides an "intelligent" interface supporting expert physicians in the acquisition of a guideline, relying on different forms of consistency checking.

Name and range checking is automatically triggered whenever the expert physician introduces a new term or value within the description of an action in a guideline, and forces her/him to use only terms/values that have already been defined within the Clinical DB. Whenever the expert physician introduces a node or arc, different controls are automatically activated to check whether the new element is consistent with several *logical design criteria*. For example, alternative arcs may only exit from a decision action. Finally, a "semantic" checking regards the *consistency of temporal constraints* in the guideline. This checking is automatically triggered whenever the expert physician saves a guideline. In fact, alternative sequences of actions and sub-actions 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 verify consistency.

While GLARE provides users with an *interface* high-level language to express temporal constraints, the temporal reasoning facility maintains a homogeneous *internal* representation of such constraints, on which the temporal reasoning algorithms operate. We based the design of the internal representation formalism on the "classical" *bounds on differences* approach and on the STP (Simple Temporal Problem) framework [1]. This framework takes into account conjunctions (sets) of bounds on the distance between pairs of time points (of the form $c \leq P1 - P2 \leq d$), and has very nice computational properties: correct and complete temporal reasoning (e.g., for consistency checking) can be performed in cubic time by a classical all-to-all-shortest-paths algorithm (such as Floyd-Warshall's one), which also provides the *minimal network* of the temporal constraints [1].

Most of the temporal constraints provided by GLARE's interface formalism can be easily represented by the STP framework. Each action in a guideline (including composite actions) can be represented by its starting and its ending point. Thus, the duration of an action can be modeled as the distance between its endpoints. Delays are directly modeled as distances between points, as well as qualitative temporal constraints. Unfortunately, the STP framework must be significantly extended if one wishes to deal with *repetitions*. We propose to represent the constraints regarding repetitions into separate STP frameworks, one for each repeated action. Thus, in GLARE, the overall set of constraints in a guideline is represented by a *tree of STP frameworks* (*STP-tree* henceforth). The root of the tree is the STP which homogeneously represents the constraints between all the actions (composite and atomic) in the guideline, except repeated actions (which are composite actions, by our definition). Each node in the STP-tree is an STP, and has as many children as the number of repeated actions it contains. Each arc in the

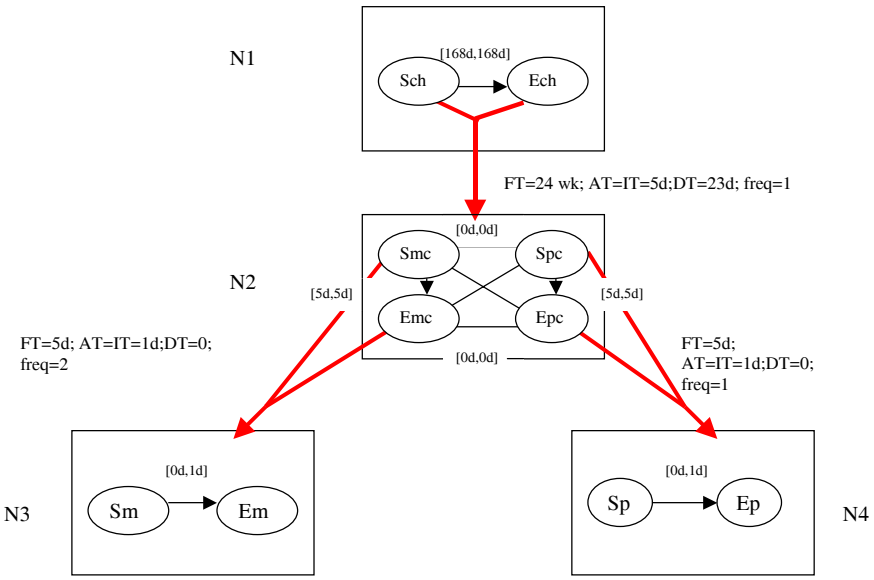

Fig. 2. STP-tree for the multiple mieloma chemotherapy guideline. Arcs between nodes in a STP are labeled by a pair $[n, m]$ representing the minimum and maximum distance between them. Arcs from a pair of nodes to a child STP represent repetitions.

tree connects a pair of points in an STP (the starting and ending point of a repeated action) to the STP containing the constraints between the related subactions, and is labeled with the list of properties describing the temporal constraints on the repetitions (AT, DT etc.; see Ex.2 below). Figure 2 shows the STP-tree representing the temporal constraints in Ex.2.

Ex.2 *One possible therapy for multiple mieloma is made by six cycles of 5-day treatment, each one followed by a delay of* 23 *days (for a total* FT *of* 24 *weeks, divided into six repetitions of an* AT *of* 5 *days, followed by a* DT *of* 23 *days. The overall therapy is reported as the root of the STP-tree in figure 2). Within each 5-day cycle,* 2 *inner cycles can be distinguished: the melphalan treatment, to be provided twice a day (*AT=IT*), for each of the* 5 *days (*FT*), and the prednisone treatment, to be provided once a day, for each of the* 5 *days. These two treatments must be performed in parallel (see the temporal constraints in node* $N2$ *in figure 2), and are shown as leaves of the STP-tree (nodes* $N3$ *and* $N4$ *respectively).*

Temporal consistency checking proceeds in a top-down fashion, starting from the root node of the STP-tree. The root is a "standard" STP, so that Floyd-Warshall's algorithm can be applied. Then, we proceed towards the leaves of the tree. For each node in the tree other than the root, we apply $ALGO1$ (see [18] for more details):

$ALGO1$**: temporal consistency of guidelines**

1. the consistency of the constraints used to specify the repetition taken in isolation is checked;

2. the "extra" temporal constraints regarding the repetition are mapped onto bounds on difference constraints;
3. Floyd-Warshall's algorithm is applied to the constraints in the STP plus the "extra" bounds on difference constraints determined at step 2.

Property 1. $ALGO1$ is correct, complete, and tractable (since it operates in O(N^3), where N is the number of actions in the guideline).

5 Execution

The typical use of our execution tool is "on-line": a user physician executes a guideline applied to a specific patient (i.e., s/he instantiates a general guideline considering the data of a given patient). However, we also envision the possibility of adopting our execution tool for "off-line" execution (this might be useful in different tasks, including education, critiquing and evaluation). In both cases, temporal reasoning and decision support facilities may be resorted to (see next subsections).

5.1 Temporal Reasoning Facilities

The execution tool exploits temporal consistency checking as well. Each action in a guideline represents a *class* (set) of instances of actions, in the sense that it will have specific instantiations for specific executions of the guideline itself. When a guideline is executed on a specific patient, specific *instances* of such actions are performed at specific times. We suppose that the exact times of all the actions in the guideline which have been executed are given in input to our system. Thus, we have to check that they respect (i.e., are consistent with) the temporal constraints they inherit from the classes in the general guideline. Moreover, also the (implicit) temporal constraints conveyed by the has-part relations between actions in the guideline must be respected, as well as those involved by periodicity and repetitions.

In a broad sense, periodic events are special kinds of classes of events, i.e., classes whose instances must respect a periodic temporal pattern. However, while inheritance of constraints about duration, delays and ordering regards single instances (duration) or pairs of instances (delays, precedence), periodicity constraints concern whole sets of instances, imposing constraints on their cardinality and on the temporal pattern they have to respect. Finally, notice that the interplay between part-of relations and periodic events might be quite complex to represent and manage. In fact, in the case of a composite periodic action, the temporal pattern regards the components, which may be, recursively, composite and/or periodic actions (see Ex.2).

Finally, notice that, when considering instances, one should also take into account the fact that guidelines have a "predictive" role. E.g., if one has observed a given action $E1$ which is an instance of a class of actions E in a guideline, and the class E' follows E in the guideline itself, one expects to observe an instance of E' in a time consistent with the temporal constraints between the classes E and E'. We assume that, as regards the treatment of hospitalized patients, we have *complete observability*, i.e., that each execution of an action of the guideline is reported in the clinical record of

the patient, together with its time of occurrence. Thus the consistency checking must consider "prediction", since not having observed an instance of an action indicates an inconsistency, unless the temporal constraints impose that it may also be executed in a time after NOW.

Our temporal reasoning algorithm can be schematized as follows:

$ALGO2$: temporal consistency on guidelines' execution

1. the existence of non-observed instances whose occurrence is predicted by the guideline is hypothesized;
2. all the constraints in the general guidelines are inherited by the corresponding instances (considering both observed and hypothesized instances). This step also involves "non-standard" inheritance of constraints about periodicity;
3. constraint propagation is performed on the resulting set of constraints on instances (via Floyd-Warshall's algorithm), to check the consistency of the given and the inherited constraints;
4. if constraints at step 3 are consistent, it is further checked that such constraints do not imply that any of the "hypothesized" instances should have started before NOW.

Property 2. Our consistency checking algorithm $ALGO2$ is correct, complete, and tractable (since it operates in $O((N+M)^3)$, where N is the number of actions in the guideline and M the number of instances of actions which have been executed.

A detailed analysis of our temporal reasoning algorithm, and of Property 2 is outside the goals of this paper, and can be found in [17].

5.2 Hypothetical Reasoning Facility

GLARE's execution tool also incorporates a decision support facility (called *hypothetical reasoning*), able to assist physicians in choosing among different therapeutic or diagnostic alternatives. The default execution of decision actions works as follows. As regards *diagnostic decisions*, the execution module automatically retrieves the parameter values from the Patient DB, evaluates the scores for every alternative diagnosis, and then compares them with the corresponding threshold. All alternative diagnoses are then shown to the user-physician, together with their scores and the threshold, and the tool lets the user choose among them (a warning is given if the user chooses a diagnosis which does not exceed the threshold). The execution of a *therapeutic decision* simply consists in presenting the effectiveness, cost, side-effects, compliance, and duration of each alternative to the physician, thus allowing her/him to select one of them. On the other hand, through the adoption of the hypothetical reasoning facility, it is possible to compare different paths in the guideline, by simulating what could happen if a certain choice was made. In particular, users are helped in gathering various types of information, needed to discriminate among alternatives. As a matter of fact, in many cases, therapeutic and/or diagnostic decisions should not be taken on the basis of "local information" alone, i.e. by considering just the decision criteria associated with the specific decision action at hand, but one should also take into account information stemming from relevant alternative paths. In particular, the resources needed to perform all

the actions found along each alternative path (starting from the decision at hand), the costs and the times required to complete them, are meaningful selection parameters. The unique feature of this tool is its capability of retrieving such "global information". This facility can be used both in the on-line and in the off-line execution mode.

Technically speaking, to provide a projection of what could happen in the rest of the guideline in case the user selected a given alternative, the tool works as follows. Through the execution tool graphical interface, the physician is asked to indicate on the graph the starting node (normally the decision at hand) of the paths to be compared and (optionally) the ending nodes (otherwise all possible paths exiting the starting node will be taken into consideration). Relevant decision parameters (costs, resources, times) will be gathered from the selected portions of the guideline in a semi-automatic way. In particular, whenever a decision action is reached within each path, the user is allowed to choose a subset of alternatives, by checking the corresponding buttons in a pop up window. For a diagnostic decision, s/he may want to allow all alternatives to be considered, or s/he could limit the search to the diagnoses that obtained a score exceeding the threshold, or to a subset of these diagnoses themselves. When dealing with a therapeutic action, again the user could allow all alternatives to be evaluated, or could mark the therapies s/he expects to be equivalent for the patient under examination, or a subset of them. Making a restriction means that, on the physician's opinion, the other paths are not interesting for comparison, and they will be ignored by the hypothetical reasoning process. If a composite action is found, it is expanded in its components, and the hypothetical reasoning facility is recursively applied to each of them, by analyzing all the decision actions that appear at the various decomposition levels. At the end of this process, the tool displays the values of the collected parameters for each one of the selected paths. The final decision is then left to the physician. Note that while resources in a path are simply listed, and costs are summed up (in the case that an exit condition is specified, the cost of each iteration will be calculated), the temporal constraint propagation techniques discussed so far are necessary in order to deal with the temporal parameters.

6 Testing

We have already tested our prototype acquisition and representation system considering different domains, including bladder cancer, reflux esophagitis and heart failure. In the case of bladder cancer, the expert physicians started designing the guideline algorithm from scratch, directly using our acquisition tool (after a brief training session), and exploiting the facilities (e.g., consistency checking) it provides.

In the cases of reflux esophagitis and heart failure, the physicians started with guideline algorithms previously described on paper (using drawings and text), and used our acquisition tool to introduce them into a computer format. The acquisition of a clinical guideline using our system was reasonably fast (e.g., the acquisition of the guideline on heart failure required 3 days).

In all the tests, our representation formalism (and the acquisition tool) proved to be expressive enough to cover the clinical algorithms.

7 Comparisons and Conclusions

In this paper, we highlighted the most innovative features of GLARE, a domain - independent framework to acquire, represent and execute clinical guidelines. In the latest years, many approaches agreed that providing a semi-automatic treatment of clinical guidelines is very advantageous, and that AI techniques can be fruitfully applied to achieve such a goal.

Among the approaches in the literature, we think that PROforma [4] and Asbru [15] are the closest ones to GLARE. However, two distinguishing features of the GLARE approach, that clearly highlight the advantages of applying AI techniques to clinical guideline tools, can be outlined:

(i) GLARE provides "intelligent" mechanisms for consistency checking (see however [13], where the correctness and completeness of the activation conditions of subtasks are automatically checked). Specific attention is devoted to the treatment of temporal constraints between atomic, periodic and/or repeated actions in both the acquisition (section 4) and in the execution (section 5) phases;
(ii) GLARE provides user physicians with the hypothetical reasoning facility, a practical way of comparing alternative paths in a guideline on the basis of a chosen set of parameters (see section 5).

In particular, (i) and (ii) are also the innovative features of the approach we described in this paper (together with the introduction of the "intermediate" XML layer in the architecture; see section 2) with respect to our initial approach, as described in [20].

More generally, GLARE, as well as PROforma, Asbru, and many other approaches, shows that the adoption of AI techniques can provide relevant advantages in the (semi-)automatic treatment of clinical guidelines, especially regarding the user - friendliness of the tools being built. In turn, user-friendliness seems to be one of the most crucial aspects in the dissemination and actual adoption of computer science tools within the medical community.

References

1. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. In R. J. Brachman, H. J. Levesque, and R. Reiter, editors, *Knowledge Representation*, pages 61–95. MIT Press, London, 1991.
2. D.B. Fridsma (Guest Ed.). Special issue on workflow management and clinical guidelines. *Journal of the American Medical Informatics Association*, 1(22):1–80, 2001.
3. F. Van Eynde. Iteration, habituality and verb form semantics. In *Proc. of third Conference of European Chapter of the Association for Computational Linguistics*, pages 270–277, 1987.
4. J. Fox, N. Johns, A. Rahmanzadeh, and R. Thomson. Disseminating medical knowledge: the proforma approach. *Artificial Intelligence in Medicine*, 14(1-2):157–182, 1998.
5. C. Gordon. Practice guidelines and healthcare telematics: towards an alliance. In C. Gordon and J.P. Christensen, editors, *Health Telematics for Clinical Guidelines and Protocols*, pages 3–15. IOS Press, Amsterdam, 1995.
6. C. Gordon and J.P. Christensen, editors. *Health Telematics for Clinical Guidelines and Protocols*. IOS Press, Amsterdam, 1995.

7. J.M. Grimshaw and I.T. Russel. Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluation. *Lancet*, 342:1317–1322, 1993.
8. G. Molino. From clinical guidelines to decision support. In *Proc. Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, LNAI 1620*, pages 3–12, 1999.
9. M.A. Musen, C.W. Carlson, L.M. Fagan, S.C. Deresinski, and E.H. Shortliffe. Automated support for community-based clinical research. In *Proc. 16th Annual Symposium on Computer Applications in Medical Care*, pages 719–723, 1992.
10. M.A. Musen, S.W. Tu, A.K. Das, and Y. Shahar. Eon: a component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Informatics Association*, 6(3):367–388, 1996.
11. M. Peleg and et al A.A. Boxawala. Glif3: The evolution of a guideline representation format. In *Proc. AMIA'00*, pages 645–649, 2000.
12. I. Purves. Computerised guidelines in primary health care: Reflections and implications. In C. Gordon and J.P. Christensen, editors, *Health Telematics for Clinical Guidelines and Protocols*, pages 57–74. IOS Press, Amsterdam, 1995.
13. S. Quaglini, L. Dazzi, L. Gatti, M. Stefanelli, C. Fassino, and C. Tondini. Supporting tools for guideline development and dissemination. *Artificial Intelligence in Medicine*, 2(14):119–137, 1998.
14. S. Quaglini, M. Stefanelli, A. Cavallini, G. Miceli, C. Fassino, and C. Mossa. Guideline-based careflow systems. *Artificial Intelligence in Medicine*, 1(20):5–22, 2000.
15. Y. Shahar, S. Mirksch, and P. Johnson. The asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14(1-2):29–51, 1998.
16. 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. *Journal of the American Medical Informatics Association*, 5(7):488–498, 2000.
17. P. Terenziani and L. Anselma. Towards a temporal reasoning approach dealing with instance-of, part-of, and periodicity. Accepted for publication in Proc. TIME'03. IEEE Press, 2003.
18. P. Terenziani, C. Carlini, and S. Montani. Towards a comprehensive treatment of temporal constraints in clinical guidelines. In *Proc. TIME'02*, pages 20–27. IEEE Press, 2002.
19. P. Terenziani, F. Mastromonaco, G. Molino, and M. Torchio. Executing clinical guidelines: temporal issues. In *Proc. AMIA'00*, pages 848–852, 2000.
20. P. Terenziani, G. Molino, and M. Torchio. A modular approach for representing and executing clinical guidelines. *Artificial Intelligence in Medicine*, 23(3):249–276, 2001.
21. P. Terenziani, S. Montani, A. Bottrighi, G. Molino, and M. Torchio. Supporting physicians in taking decisions in clinical guidelines: the glare's "what if" facility. *Journal of the American Medical Informatics Association, Proc. Annual Fall Symposium*, 2002.
22. S.W. Tu, M.G. Kahn, M.G. Musen, J.K. Ferguson, E.H. Shortliffe, and L.M. Fagan. Episodic skeletal-plan refinement on temporal data. *Comm. ACM*, 32(12):1439–1455, 1989.
23. L. Vila. A survey on temporal reasoning in artificial intelligence. *AI Communications*, 1(7):4–28, 1994.