Towards a Comprehensive Treatment of Temporal Constraints in Clinical Guidelines

Paolo Terenziani, Carlo Carlini, Stefania Montani
DISTA, Univ. del Piemonte Orientale “Amedeo Avogadro”
Corso Borsalino 54, Alessandria, Italy
Phone: +39 0131 287447 E-mail: {terenz,stefania}@mfn.unipmn.it

Abstract

In this paper, we focus on an application and extension of Artificial Intelligence temporal reasoning techniques in order to represent and reason with temporal constraints in clinical guidelines. Particular attention is dedicated to the treatment of repeated (periodic) events, which play a major role in clinical therapies. We also discuss some limitations of our current approach, highlighting possible future enhancements. The work reported in this paper has been developed within the GLARE project, meant to realize a prototype of a domain-independent manager of clinical guidelines. The GLARE system has been built in cooperation with Azienda Ospedaliera S. Giovanni Battista of Turin, and has been successfully tested on different clinical domains.

KEYWORDS: clinical guidelines; temporal constraints; repeated/periodic events; consistency checking; query answering

1 Introduction

Basically, clinical guidelines are a means for specifying the “best” clinical procedures and for standardizing them. In the latest years, the medical community has started to recognize that a computer-based treatment of clinical guidelines provides relevant advantages (see, e.g., [1,2]). Thus, many different approaches have been developed to achieve such a goal (see, e.g., the collections [3,4]). A main issue in developing computer-based tools dealing with guidelines is the treatment of temporal phenomena. For instance, Shahar has proposed a comprehensive approach dealing with temporal abstraction within the clinical procedures [5]. Miksch has foused on a user-friendly tool to show the temporal constraints in a guideline to physicians in a graphical and commonsense way [6]. On the other hand, to the best of our knowledge, no approach has focused specifically on temporal reasoning about temporal constraints within the actions in a guideline. Temporal reasoning can be useful both when the guideline is being acquired (e.g., to check the consistency of the constraints imposed between actions by the expert physicians), and when it is executed on specific patients (e.g., to look for temporally minimal procedures to deal with a given situation). When dealing with real-world clinical guidelines, different types of explicit and implicit temporal constraints have to be managed. For example, the *part-of* relations between clinical actions implicitly underly temporal constraints between actions and subactions. Moreover, constraints involving repeated (periodic) actions must be taken into account, besides “classical” qualitative and quantitative temporal constraints [7]. Thus, there is no Artificial Intelligence (AI) temporal reasoning tool which easily adapts in order to deal with all such types of temporal constraints. In this paper, we first describe the GLARE system (section 2; see [11-14] for more details). In section 3, we briefly analyse the types of temporal constraints that occur in clinical guidelines. Sections 4, 5 and 6 present the original contribution of the paper, proposing an *application* of STP-based temporal reasoning techniques [8-10]: in section 4 we show how a labeled tree of STP’s can be used in order to deal with all the types of constraints introduced in section 3, and in sections 5 and 6 we show how STP-based constraint propagation and query answering techniques can be extended to operate on such a tree.

2 The GLARE System

GLARE (GuideLine Acquisition, Representation and Execution) is a system dealing with clinical guidelines we developed in cooperation with Azienda Ospedaliera S. Giovanni Battista of Turin starting from 1997. In this section, we highlight some of the main features of the system (see [11-14] for more details). GLARE is a *domain-independent* system, since, in principle, it can deal with clinical guidelines in any clinical domain. GLARE distinguishes between the

acquisition phase (when a guideline is introduced into the system –e.g., by a comettee of expert physicians) and the ***execution*** phase, when a guideline is applied to a specific situation (i.e., it is instantiated on a given patient). The system architecture is therefore composed by two main tools, the *acquisition module* and the *execution module*. The tools strictly interact with a set of databases, including the *clinical* database, which provides physicians with a standard terminology, and the *patient* database, which contains the patients' data.

2.1 Representation formalism

In the design of GLARE's **representation formalism**, we have tried to mediate between expressiveness and usability/user-friendlyness. We thus provide a *limited* set of "commonsense" primitives. We distinguish between ***atomic*** actions (elemenatry steps in a guideline) and ***composite actions*** (called ***plans***). Each plan has a starting action, and is composed by a set of sub-actions, which must be executed depending on the ***control relations*** between them. In turn, each sub-action may be atomic or composite. In our approach, the same action cannot belong to different plans. Thus, each guideline can be represented as a *tree of graphs* in which the root of the tree represents the guideline itself, each node represent an action and has as children its subactions (atomic actions are leaves); control relations are represented as arcs between nodes in each graph (see the graphical representation of one of such graphs in Figure 1). We distinguish among four types of atomic actions: ***work actions*** (basic actions, to be executed by physicians), ***query actions*** (data requests), ***decisions*** (embodying the criteria to select from alternative paths in a guideline) and ***conclusions*** (outputs of a decision). Each atomic action can be described by a set of properties including, e.g., name, textual description, cost, duration etc. [12,14].

GLARE distinguishes between four types of ***control relations*** between actions in a plan. A **sequence** relation from an action A_1 to an action A_2 represents the fact that the execution of A_2 must start after the end of A_1. (Possibly imprecise) temporal delays can specified between the end of A_1 and the start of A_2. Also *infinite delays* are coped with, to represent qualitative constraints. An **alternative** relation involves a decision action D and a set of alternative actions A_1, …, A_k. Depending on the decision action, one of the actions A_1, …., A_k will be chosen for execution. A delay between the end of the decision and the start of each alternative action can be specified. In order to cope with temporally complex cases, we introduced the "**controlled**" relation. It applies to a set A_1, …, A_k ($k>1$) of actions, and can be used to specify minimum and maximum delays between any pair of starting/ending points of such actions. Finally, the **repetition** relation apply to a plan A (by definition, we say that each repeated action is a plan) and allows one to specify constraints on the number of repetitions of A and on their periodicity (see section 3).

2.2 Acquisition module

GLARE's acquisition module provides expert-physicians with a user-friendly and easy-to-use tool for acquiring clinical guidelines. It provides: (i) a graphical interface, which supports primitives for drawing the control relations between actions, and ad hoc windows to acquire their properties (e.g., costs); (ii) facilities for browsing the guideline; (iii) "intelligent" help and consistency checking. As regards issues (i) and (ii), figure 1 shows part of our guideline for Gastro Esophageal Reflux Disease (henceforth called GERD). The left part of the figure displays the window representing the general structure of the guideline: it shows in form of a tree the *part-of* hierarchy of actions introduced by the expert physician and can be used for browsing the guidelines. Each node represents an action, and each action has as children the sub-actions composing it. The right part of the figure shows the window used to acquire the control relations between the components of composite actions. Each sub-action is represented as a node in the graph (different forms and colours are used to distinguish among different types of actions), while control relations are represented as arcs. In the example, we show the top-level actions of GERD. By clicking on the nodes in the graph, the user can trigger other windows used in order to acquire the internal descriptions (attributes) of nodes. As regards point (iii), n*ame 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, by forcing her/him to use only terms/values that have already been defined within the *clinical* database. Whenever the expert physician introduces a node or arc within a guideline, 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" check regards the *consistency of temporal constraints* in the guideline (see section 3).

2.3 Execution

The execution of actions depends on their type, and is based on an *"agenda technique"* [13,14], being the agenda a data structure meant to contain the next action to be executed. For example, *work actions* are simply pushed onto the agenda, until the user physician completes them and deletes them from the agenda itself; on the other hand, the execution of a *diagnostic decision* involves automatically evaluating the scores for each alternative diagnosis (on the basis of the patient's data in the *patient* database) and showing them to the user-physician, who *choosse* among them. At any stage in the execution process, the user-physician can suspend the execution and take advantage of the *hypothetical reasoning* support provided by GLARE. This form of reasoning provides a simulation of what could happen in case user-physicians selected a given alternative. Basically, it allows one to compare different paths in the guideline, according to pre-defined sets of parameters (currently: resources, costs, time). A graphical interface lets the user specify the parameters chosen for the comparison, the starting node of the paths to be compared and (optionally) the ending nodes. Within each path, whenever a decision action is reached, the user may select a subset of alternatives. At the end of the process, the tool displays the values of the chosen parameters for each of the selected paths. While resources in a path are simply collected, and costs are summed up, temporal constraint propagation techniques are necessary in order to deal with the temporal parameters (see section 5).

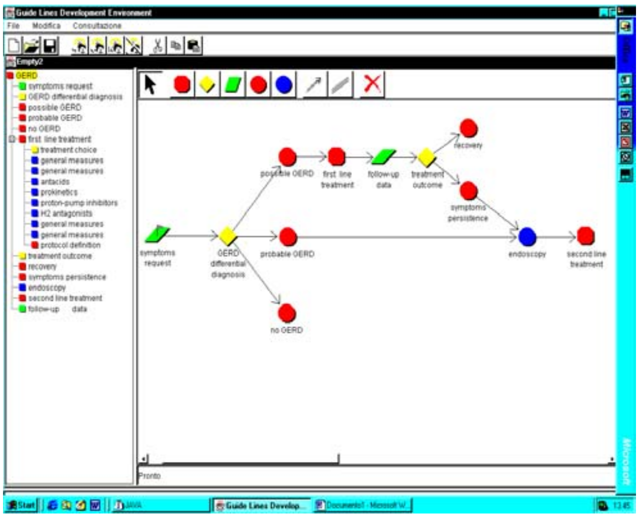

Figure 1. Representation of the top level of the guideline about GERD, as it is shown by the graphical interface of GLARE's acquisition module.

2.4 Testing

We have already tested our prototype acquisition and representation system considering different domains, including bladder cancer, reflux esophagitis, and heart failure. In all the tests, our representation formalism proved to be expressive enough to cover the clinical algorithms, and the acquisition of a clinical guideline algorithm by expert physicians was reasonably fast (e.g., the acquisition of the guideline on heart failure required 3 days).

3 Temporal constraints within clinical guidelines

Three different types of temporal constraints are involved within clinical guidelines dealt with by the GLARE system:

(i) "standard" distance constraints between actions belonging to the same plan. These constraints are expressed within the sequence, alternative and controlled relations between actions. GLARE also allows one to specify the minimum and maximum duration of atomic actions;

(ii) implicit constraints involved by the *part-of* relations between actions. In particular, the starting point of a plan must coincide with the starting point of its first node. Moreover, the minimum and maximum duration of a plan must correspond to the duration of the temporally shortest and longest alternative chains of actions in the plan;

(iii) constraints about repetition of actions, and, possibly, about their periodicity. Our formalism supports two different ways of specifying repetitions (which can be combined). One way is to state that the action has to be performed until a given **exit condition** becomes true (at execution time, the exit condition is tested before each repetition.). The second way is to specify a duration (**frame-time**) for the repetitions. Currently, only *exact* frame times are supported. In both cases, the physician needs to specify the prototypical repetition. This involves specifying the **frequency** of the repetitions in time. In general, specifying frequencies requires a 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** (2 in the example) in order to define the periodicity of the action. Moreover, the **number of executions** in the given periodicity must also be indicated (3 in the example).

4 Representing temporal constraints in clinical guidelines

In this section, we first briefly sketch the bounds on differences constraints and the STP framework [8], and then we show incrementally how this framework can be extended in order to cope with the temporal constraints pointed out in section 3.

4.1 Bounds on Differences

Bounds on differences are constraints of the form $c \leq x-y \leq d$ (where c and d are real numbers), which have an intuitive temporal interpretation, namely that the temporal distance between the time points x and y is between c (minimum distance) and d. The literature about constraints in the AI field has widely shown that the distance primitive is very powerful [7,8]. In the temporal area, two main frameworks based on bounds on differences have been used. The *STP* framework only takes into account conjunctions (sets) of bounds on differences, that can be represented in a *graph* in which each node is a variable (time point) and each arc is a distance [8]. The STP framework can be used to model *precise* or *imprecise* temporal locations (dates), durations, delays between points, and different forms of qualitative temporal constraints between time points and/or time intervals (see [15,10]). The STP framework has very nice computational properties: correct and complete propagation of the constraints (e.g., for consistency checking) can be performed in cubic time, and can give as output the *minimal network* of the constraints (i.e., the minimal constraints between each pair of entities) [8]. The minimal network can be computed by an "all-to-all shortest paths" algorithm such as Floyd Warshall's one, and can also be used for efficient query answering [9] and incremental insertion of constraints [10]. In the TCSP framework, on the other hand, also disjunctions of distances are allowed. This allows for disjunctive constraints, at the price of exponential consistency checking algorithms [8].

4.2 Temporal constraints between actions belonging to the same plan

Temporal constraints between actions in a plan can be easily mapped onto a STP framework, in which each action is represented by its starting and its ending point. Durations can be represented as the distance between the starting and the ending point. As regards the control relations, the delays associated with sequence and alternative relations can be mapped as distances between points, as well as "controlled" relations. Additional bounds on difference constraints model explicitly the fact the the (starting point of the) start action of a plan precedes all the actions in the plan. Constraint propagation can be applied to such an STP framework in order to check consistency and obtain the minimal network. However, the resulting minimal network connects each pair of nodes, including nodes belonging to alternative paths (e.g., "probable GERD" and "symptoms persistence" in Figure 1). Thus, besides the minimal network, also the original guideline must be taken into account, if one has to deal with reachability between actions (e.g., at query time).

4.3 Temporal constraints in different plans

In GLARE, clinical guidelines consist also of composite actions (plans), which sub-actions may be, in turn, composite. This means that our formalism supports a hierarchy of actions related by "part-of" relations. Thus, in general, we have to cope with

(i) temporal constraints in different plans (graphs).

Moreover, besides the constraints within actions in a plan (intra-plan constraints; see subsection 4.2), we have to model also:

(ii) temporal costraints involving (the endpoints of) plans, and

(iii) temporal constraints between each plan and its subactions.

From the temporal point of view, we represent a plan via its starting and its ending node. In such a way, also the temporal constraints (ii) above, i.e., the constraints between a plan and the other actions (atomic actions and/or plans) at the same hierarchical level (i.e., which are part of the same plan; consider, e.g., "first line treatment" and "possible GERD" in figure 1) can be simply represented as discussed in subsection 4.2. Temporal constraints of type (iii) above are represented as constraints between the starting/ending points of the plan and the endpoints of its (direct) subactions. In particular, the starting point of a plan is temporally equal to the starting point of its starting action, while its ending point is after or equal to the ending point of the last action in each alternative path of the sub-actions composing it[1]. In such a way, all the constraints involving actions and plans (which are *not repeated*)

[1] For instance, the ending point of the GERD plan is after or equal than the ending points of "recovery", "second line treatment" and "no GERD"; see figure 1. Notice that, in the case a path ends with a set actions which are in a "controlled" relation, the less or equal constraint is imposed on all such actions, since all the actions might potentially be the last one in the path –depending on the constraints. Finally, notice also that, in the current version, we do not deal with the strictest constraint that at least one of the ending points of the last actions in each alternative path is temporally equal to the ending point of the plan containing them; in fact, such a disjunctive constraint cannot be modeled in the STP framework.

can be homogeneously represented into a unique STP framework, containing a starting and an ending points for each plan and atomic action in the guideline[2].

4.4 Temporal constraints about repeated actions

The above homogeneous approach must be significantly extended if one wants to deal with repeated actions. Let us suppose, for instance, that a repeated plan *P* is composed by a sequence of two subactions A_1 and A_2, which must be repeated (1) until a given condition *C* is satisfied, or (2) once a day, for 2 months. In both cases, stating that the starting point of *P* coincides with the starting point of A_1, and *P*'s ending point is less or equal to A_2's ending point could lead to temporal inconsistencies. In fact, what we actually need to represent is the fact that *P*'s starting point coincides with the starting point of *the first repetition* of A_1, and *P*'s ending point to the ending point of the *last repetition* of A_2. Such constraints cannot be modeled into the homogeneous STP framework discussed above[3].

We thus chose to represent the constraints regarding repeated actions into separate STP frameworks, one for each reapeated plan. For example, the (very simple) example above is modeled via the introduction of a separate STP framework representing the constraints between A_1's and A_2's endpoints. Such an STP is related to the endpoints of *P* via an arc which is labelled with the temporal properties of the repetition. Thus, in GLARE, the overall set of constraints in a guideline is represented by a ***tree of STP frameworks*** (*STPs-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 plans, by our definition). Each node in the tree is an STP, and has as many children as the number of repeated actions it contains. Each arc in the tree connects a pair of endpoints in an STP (the starting and ending point of a repeated action) to the STP containing the constraints between its subactions, and is labeled with a list of properties describing the temporal constraints on the repetitions (e.g., frame time, granularity,… see section 3). Figure 2 shows a graphical representation of a tree of STPs including our simple example (consider node STP_k).

5 Reasoning with temporal constraints in clinical guidelines

Temporal reasoning can be used in the acquisition phase, in order to check the consistency of the temporal constraints being introduced. Moreover, it can also be used at execution time, as discussed in the following section.

In GLARE, we organised the temporal constraints in order to exploit the STP framework's reasoning facilities as much as possible. However, in the STPs-tree, temporal information is not only conveyed by the bounds on differences in each STP, but also by the temporal properties on the arcs of the tree. Temporal consistency checking can thus proceed in a top down fashion, starting from the root of the STPs-tree. Basically, the root contains a "standard" STP, so that Floyd-Warshall's all-to-all shortest path standard algorithm can be applied to check its consistency, and to produce the minimal network of the constraints it contains (*local minimal network* henceforth). Then, we proceed top down towards the leaves of the tree. For each node N_i in the tree (except the root), we operate in three steps:

(1) first, we check the consistency of the constraints used to specify the repetition (i.e., the constraints on the arc entering N_i) taken in isolation;

(2) second, we translate the temporal constraints on the entering arc into bounds on differences constraints on the appropriate time points in the STP contained by N_i;

(3) third, we apply Floyd-Warshall algorithm to the constraints in N_i plus the "extra" bounds on

[2] Alternatively, we could have chosen to partition the constraints into a set of STP frameworks, one for each plan in the guideline (e.g., one STP could be used to model the constraints between the actions in figure 1 – which are the subations of the GERD plan –, another STP to model the constraints between the subactions of the "first line treatment" plan – see figure 1–, and so on). Such a clustering technique would save space and, apparently, make the reasoning process more efficient. However, the (ii) and (iii) constraints above make the constraint in different STPs dependent on each other. Thus, determining the global consistency of such a set of STPs would need a potentially never-ending iteration of the reasoning process, until a state of quiescence is reached. In fact, the application to the shortest paths algorithm to one STP (say STP_1) may modify constraints which also affect points in other STPs (say STP_2). In turn, reasoning on STP_2 may impose new constraints that affect points into other STPs (say STP_3, or also STP_1), and so on.

[3] Actually, in the case (2) above, one could explicitly represent in the STP framework all the repetitions of A_1 and A_2. Given such an explicit representation, the constraints relating such repetitions to the endpoints of *P* can be homogeneously represented into a unique STP framework, along the lines shown in subsection 4.3. However, in the clinical domain (especially in the therapeutic context), actions can be repeated a relevant number of times, so that an explicit representation of all repetitions could cause an unnecessary explosion of the number of points in the STP. Moreover, such an "explicitation-based" technique cannot be applied to the case (1) above, in which the number of repetitions is not known a priori (it can only be determined at execution time).

differences constraints determined at the previous step. This also provides a local minimal network for the above constraints only.

While the third step is trivial, the first two steps are performed by ad-hoc specialised algorithms. In the first step, the algorithms looks for the consistency of the different components in the specification of repeated plans. For instance, the frequency must be contained into the frame time.

Also the second step is performed by an ad-hoc algorithm, that extracts the information concerning the maximum duration of the repeated action *RA* which constraints are represented by N_i, and imposes such constraints between the starting point of the first action composing RA (in N_i) and the ending points of each one of the possible last actions (the last actions of each alternative path which is part of the plan RA in N_i). In case the repetition has a frame time (e.g., "…for 10 days…"), its maximum duration trivially corresponds to the frame time. In case the repetition has been specified via a frequency (e.g., "… 3 times each two days …"), the maximum duration is given by the difference between the product of grouping and granularity (2 days in the example) and the product of the minimum duration of a whole repetition times the number of executions minus one. On the other hand, in case an exit condition (and no frame time) is specified, the maximum duration is infinite. The algorithm takes advantage also of a conversion table mapping each granularity provided to the user (currently, GLARE does not deal with user-defined granularities) into the basic granularity adopted by the reasoning process.

Property 1. The top down visit of the STPs-tree is complete as regards consistency checking of the constraints in the STP-tree.

Proof (sketch). The all-shortest-paths algorithm is complete for STP frameworks. The only constraints relating actions (or, better, time points) in different STPs in the tree are located on the arcs of the tree, and, by definition, the STPs-tree does not allow loops between STPs. Since frame times and/or frequencies (if any) must be provided in a precise (exact) way, they cannot be further "restricted" by constraint propagation. Thus, there is no need to propagate forward and backward the constraints along the tree, and a top-down visit of the tree is sufficient.

Notice, however, that the above reasoning mechanisms does not provide the minimal network between all the actions in the guideline (actually, a minimal network in the classical sense could only be provided in case all the executions of repeated actions are made explicit – see footnote 3). On the other hand, only *local* minimal networks, concerning all non-repeated actions (root of the STPs-tree) or the components of each repeated action (the other nodes in the STPs-tree), are evaluated and stored.

6 Query answering

The above temporal constraint propagation technique, and the local minimal networks obtained for each node in the STPs-tree can also be used in order to answer temporal queries. In particular, our *hypothetical reasoning* facility allows users to compare selected paths in the guideline, according to time and/or other parameters (see subsection 2.3). The minimal and maximum duration of a user-selected path starting from an action A_1 and ending to an action A_n can be efficiently provided taking advantage of the *local* minimal networks, as follows:

(i) if A_1 and A_n are not components of repeated actions (i.e., if both A_1's and A_n's endpoints are in the in the root node of the STPs-tree) the local minimal network of the root of the STPs-tree contains the minimal and maximum distance between the endpoints of A_1 and A_n;

(ii) if A_1, or A_n, or both, are components of repeated actions, the user also has to specify which repetition of A_1 and/or A_n s/he wants to focus on (if no specification is given, the first repetition is assumed by default). This operation must be iterated in the case of nested repetitions[4]. Then, the distance between A_1 and A_n is evaluated on the basis of the path on the STPs-tree corresponding to the path selected by the user in the guideline. For each node of the STPs-tree different from the lowest common ancestor in the tree, the minimal network is used in order to look for (1) the duration of a whole repetition and (2) the distance between the endpoint of the reference action (which is initially A_1) and the end of one complete repetition. (1) and (2), together with the frame time or with the frequency of repetitions are then used in order to determine the temporal distance between the reference action and the end of the repeated plan containing it. This operation is iterated to the mother nodes in the STPs-tree until the node corresponding to the lowest common ancestor of A_1 and A_n is reached. Analogous operations are applied to A_n (considering the distance between the starting actions of repetitions and the reference action), and, finally, all the distances are summed up.

[4] E.g., if A_1 is part of the repeated action A'_1 which is part of the repeated action A''_1, the user can for example refer to the second execution of A_1 within the third execution of A'_1 within the second execution of A''_1, and so on.

Case (i) above just requires constant time (a lookup in the root's local minimal network). Moreover, taking advantage of the local minimal networks, also the operations described at point (ii) above only require constant time for each visited node in the STPs-tree. In both cases, Property 2 holds.

Property 2. The query answering algorithm we provide is complete, in the sense that it generates the same answer one could have by explicitly representing the endpoints of all the actions, including all the executions of each repeated action in a global STP (see, however, footnote 3), and then computing the overall minimal network of the constraints.

Finally, the "local" minimal networks in the STPs-tree can also be used for answering efficiently yes/no queries, such as "Is it necessary/possible that the temporal distance between A1 and A2 is between c1 and c2, and/or the temporal distance between A3 and A4 is between c3 and c4?".

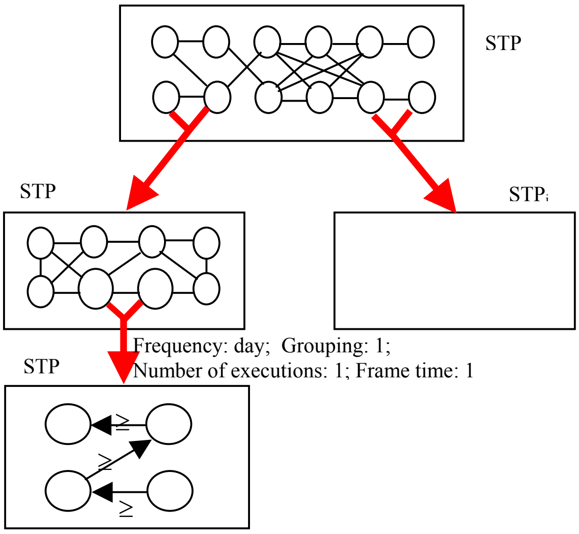
STP
STP
STPi
Frequency: day; Grouping: 1;
Number of executions: 1; Frame time: 1
STP
≥
≥
≥

Figure 2. Tree of STP frameworks: a naïve graphical representation. Thiny lines and arcs between nodes in a STP represent bound on differences constraints. Arcs from a pair of nodes to a child STP represent repetitions. E.g.,, P (whose starting point and ending point are represented by P_S and P_E respectively) is a repeated action composed by A1 and A2, and A1 precedes A2.

7 Comparisons and future work

In this paper, we sketched an *application* of STP-based temporal reasoning techniques to the treatment of temporal constraints in clinical guidelines. In particular, we focused on the extensions needed in order to deal with the implicit temporal constraints involved by *part-of* relations between actions, and by the explicit constraints regarding the (possibly periodic) repetition of actions.

Although different approaches to clinical guidelines focused on temporal problems (consider, e.g., [5,6,16,17]), no explicit treatment of the above issues is present in the literature we are aware of. On the other hand, in planning, many approaches focused on the treatment of temporal constraints on pre-defined (possibly alternative) plans (see, e.g., [18,19]). However, only little and ad-hoc work regards also (temporal constraints about) repeated events.

The approach described in this paper is part of the GLARE system. Although our temporal approach to clinical guidelines is significantly more powerful (at least from the *temporal reasoning* point of view) than the other approaches in the literature, and proved to be effective on tests in different clinical domains, further extensions would be needed. In particular, our adoption of the STP framework provided relevant computational advantages. However, a TCSP-based approach [8] could also allow us to cope with disjunctive temporal constraints (such as the constraint pointed out in footnote 1, or constraints imposing mutual exclusion between actions in a "controlled" relation), at the price of exponential complete consistency checking. Another extension could involve the possibility of futher increasing the expressiveness of the language to specify repetitions, e.g. allowing one to specify the number and the temporal location of repetitions in a possibly imprecise way. Finally, in the last decade, many approaches focused on the treatment of *user-defined* periodicities (see, e.g., [20,21] and the survey in [22]) and on temporal constraints based on user-defined periodicities [23] ([23] also presents a survey of many different AI and Temporal Data Bases (TDB) approaches to temporal constraints about periodic events). We are currently investigating the impact of integrating such work in the GLARE system. However, we are carefully studying such extensions, since we would like to retain both *completeness* and *tractability* of the consistency checking and query aswering processes, which are quite strong desiderata for the physicians we are working with.

Acknowledgements

Part of the work described in this paper has been developed within the project "Metodologie informatiche innovative per la gestione computerizzata del paziente ed il controllo di qualita'" supported by a grant of the Azienda Ospedaliera S. Giovanni Battista of

Turin. The authors are very grateful to Prof. Gianpaolo Molino and Prof. Mauro Torchio of the above Azienda for their suggestions, and their continuous and fruitful cooperation in the project.

References

[1] J.M. Grimshaw, I.T. Russel, Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluation. *Lancet*, **342-342**, 1317-1322, (1993).

[2] 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, 1995) 57-74.

[3] C. Gordon, Practice Guidelines and healthcare telematics: towards an alliance, in: C. Gordon and J.P. Christensen, eds., *Health Telematics for Clinical Guidelines and Protocols* (IOS Press, Amsterdam, 1995) 3-15.

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

[5] Y. Shahar, A Framework for Knowledge-Based Temporal Abstraction in Clinical Domains, Artificial Intelligence 90(1), 79-133 (1997).

[6] S. Miksh and R. Kosara, Communicating Time-Oriented, Skeletal Plans to Domain Experts Lucidly, Proc. *DataBase and Expert Systes Applications (DEXA 99)*, LNCS 1677, Springer Verlag, 1041-1051, 1999.

[7] Ll. Vila, "A Survey on Temporal Reasoning in Artificial Intelligence", *AI Communications 7(1)*, 4-28, 1994.

[8] R. Dechter, I. Meiri, and J. Pearl, "Temporal Constraint Networks". *Artificial Intelligence 49*, 61-95, 1991.

9] V. Brusoni, L. Console, and P. Terenziani, On the computational complexity of querying bounds on differences constraints, *Artificial Intelligence*, **74(2)**, 367-379, (1995).

[10] L. Console, P. Terenziani. Efficient processing of queries and assertions about qualitative and quantitative temporal constraints. Computational Intelligence 15(4), 442-465, November 1999.

[11] A. Guarnero, M. Marzuoli, G. Molino, P. Terenziani, M. Torchio, K. Vanni. Contextual and Temporal Clinical Guidelines, in Proc. AMIA'98 (1998) 683-687.

[12] P. Terenziani, P. Raviola, O. Bruschi, M. Torchio, M. Marzuoli, G. Molino, Representing Knowledge Levels in Clinical Guidelines, in: Proc. *Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, LNAI 1620 (Aalborg, Denmark, 1999) 254-258.

[13] P. Terenziani, F. Mastromonaco, G. Molino, M. Torchio, Executing clinical guidelines: temporal issues, Proc. AMIA'00, (2000), 848-852.

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

[15] I. Meiri, "Combining Qualitative and Quantitative Constraints in Temporal Reasoning", *Proc. AAAI*, pp. 260-267, 1991.

[16] Y. Shahar, S. Mirksch, 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).

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

[18] J. Allen, Planning as Temporal Reasoning, Proc. *Knowledge Represenation*, 3-14, 1991.

[19] F. Song and R. Cohen, Temporal Reasoning During Plan recognition, Proc. *AAAI'91*, Vol.1, 247-252, 1991.

[20] F. Kabanza, J.-M. Stevenne, and P. Wolper, "Handling Infinite Temporal Data", *Journal of Computer and System Sciences 51*, 3-17, 1995.

[21] P. Terenziani. Symbolic User-defined Periodicity in Temporal Relational Databases. Accepted for pubblication in *IEEE Transactions on Knowledge and Data Engineering*, 2002.

[22] A. Tuzhilin and J. Clifford, "On Periodicity in Temporal Databases", *Information Systems,* vol. 20, no.8, pp. 619-639, December 1995.

[23] P. Terenziani, Integrating calendar-dates and qualitative temporal constraints in the treatment of periodic events. *IEEE Transactions on Knowledge and Data Engeneering* **9(5)**, 763-783, 1997.