Automating Oncology Therapy Plans by means of Temporal Hierarchical Task Networks Planning

Juan Fdez-Olivares[1] and **Juan A. Cózar**[2] and **Luis Castillo**[3]

Abstract. This paper is focused on the representation of oncology treatment protocols by a temporally extended, Hierarchical Task Networks (HTN) based knowledge representation as well as their interpretation by a temporal HTN planning process. The planning process allows to obtain temporally annotated therapy plans that support decisions of oncologists in the area of paediatrics oncology.

1 MOTIVATION

The development of therapy planning systems [3, 16, 18] aimed to recommend predefined general courses of action to be applied to a patient, on the process of treating a disease, is an active research area in the general field of Clinical Decision Support Systems [1, 7]. Decision support systems for therapy planning incorporate, on the one hand, a computerized representation of clinical protocols, also called computer interpretable clinical guidelines (CIGs)[15]: evidence-based operating procedures that physicians follow as a guide in order to perform clinical tasks as well as making clinical decisions. Most of the approaches in this field have focused on the development of languages and frameworks to support modeling, editing and representing CIGs [12], all of them based on "Task Networks Models" [15] (the knowledge represented follows a procedural scheme based on tasks and how they are decomposed into subtasks) where mechanisms to represent workflow patterns[13] that describe the process logic between subtasks are also included (mainly sequential, conditional, iterative and synchronization control structures). On the other hand, some systems [3, 8, 17] incorporate a reasoning process that is driven by the procedural knowledge encoded in protocols and, thus, interprets such representation by supporting clinical decisions made by experts. The great part of these approaches have centered on temporal constraints reasoning [8, 17] aimed to validate constraints on a previously generated plan [18], but very little attention has been paid to the automated generation of therapy plans [4, 16].

AI Planning and Scheduling (P&S)[10] seems to be the most adequate set of techniques to cover this aspect since it deals with the development of planning systems capable of interpreting a planning domain as a set of actions schemes (that might support the representation of a clinical protocol) and reasoning about them in order to compose a suitable plan (a sequence of actions) such that its execution reaches a given goal (to treat a patient) starting from an initial state (that might represent a patient profile). Concretely, HTN planning [6, 14] becomes the most suitable AI P&S technique since it supports the modeling of planning domains in terms of a compositional hierarchy of tasks networks representing compound and primitive tasks by describing how every compound task may be decomposed into (compound/primitive) sub-tasks and the order that they must follow, by using different methods, following a reasoning process driven by the procedural knowledge encoded in its domain. These techniques have been successfully applied to real problems [9, 5] but the main criticism received, regarding their application to therapy support in the medical domain [3], has been centered in their incapacity to represent and manage crucial temporal aspects needed in this domain, as well as lack of support for a flexible execution of plans so obtained. Indeed, this has been true until very recently [6], where HTN techniques have been enhanced with valuable temporal extensions that allow to cope with a very rich temporal representation, as well as to obtain plans that could be flexibly executed as they contain temporal constraints that can be adapted during plan execution.

Therefore, in this paper we will describe an application of temporal HTN planning techniques to both, represent computer interpretable oncology clinical protocols, and automatically generate personalized therapy plans for oncology patients, following a deliberative hierarchical planning process driven by the procedural knowledge presented in such protocols. The representation language that supports the description of such knowledge also allows to represent temporal constraints that are incorporated in the reasoning process in order to obtain temporally valid plans, suitable to be applied as oncology therapy plans. Furthermore, the representation and visualization of oncology therapy plans has been developed in close collaboration with oncologists during a proof of concept of this technology in the Hospital Complex of Jaén (Spain).

2 DOMAIN OF APPLICATION

The work here presented is focused on the paediatrics oncology area, in which health assistance (and particularly therapy planning) is based on the application of oncology treatment protocols: a set of operating procedures and policies to be followed in both stages, treatment and monitoring of a patient. The main goal of an oncologist when planning a treatment is to schedule chemotherapy, radiotherapy and patient evaluation sessions. These sessions should be planned following different workflow patterns [13], included in the protocol, that specify tasks at different levels of abstraction, including sequential, conditional and iterative control flow logic constructs. Furthermore, sessions are organized as cycles of several days of duration where every cycle includes the administration of several oncology drugs. Additionally, drugs are administrated following different *administration rules* regarding their dosage and duration. Monitoring

[1] Department of Computer Science and A.I, University of Granada, Spain,email: faro@decsai.ugr.es
[2] Pediatrics Oncology Service, Hospital Complex of Jaén, Spain,email: jcozarolmo@hotmail.com
[3] Department of Computer Science and A.I, University of Granada, Spain,email: L.Castillo@decsai.ugr.es

(a)

```
(:task Protocol
 :parameters (?p - Patient ?date - Date)

 (:method Group1
  :precondition (= (group ?p) Group1)
  :tasks (
   (eval_patient ?p)
   [((and (= ?duration 360)(>= ?start ?date))
          (ChemoTherapy ?p))
       (RadioTherapy ?p)]
   (eval_patient ?p)))

 (:method Group2
  :precondition (= (group ?p) Group2)
  :tasks (
   (eval_patient ?p)
   [((and (= ?duration 360)(>= ?start ?date))
          (ChemoTherapy ?p))
       (RadioTherapy ?p)]
   (eval_patient ?p)
   [((= ?duration 360)(ChemoTherapy ?p))
       (RadioTherapy ?p)]
   (eval_patient ?p))
```

(b)

```
(:derived (patient_ok ?p)
 (and (> (leucocites ?p) 2000)
      (> (neutrophils ?p) 500) ))

(:durative-action AdminDrug
 :parameters (?p - Patient ?ph - Drug ?ds ?dur - number)
 :duration (= ?duration ?dur))
 :condition (patient_ok ?p)
 :effect (increase (total_dosage ?p ?ph) ?ds))

(:task ChemoTherapy
 :parameters (?p - Patient)

 (:method repeat
  :precondition (> (NRep ?p VCR) 0)
  :tasks (
   (:inline () (decrease (NRep ?p VCR)))
   (:inline () (assign ?dosage (* (surface ?p) (intensity ?p)))
   (:inline () (assign ?dur (* (surface ?p) (time_rate ?p))))
   (AdminDrug ?p VCR ?dosage ?dur))))

 (:method base_case
  :precondition (= (NRep ?p VCR) 0)
  :tasks ()))
```

Figure 1. HTN-PDDL concepts: (a) A *compound task* with two *decomposition methods*. (b) A *derived literal*, a *primitive task* and a task with a recursive decomposition scheme, including *inline tasks*.

sessions must also be scheduled. Therefore, in most therapies, actions concerning drugs administration and patient evaluation have to be performed according to a set of temporal constraints describing their relative order, and the delays between them. Additionally, in many cases, actions must be repeated at regular (i.e. periodic or following a repetition pattern) times. Furthermore, it is also necessary to carefully take into account the (implicit) temporal constraints derived from both, the hierarchical decomposition of actions into their components, and from the control-flow of actions in the clinical protocol [17]. All these rules, tasks and decisions vary depending on a given patient profile and may change as the treatment is going on.

At present, planning a therapy in the hospital services that concern to this work (paediatrics oncology services in the public health system of Andalusia) is done by hand, that is, thought it is possible to access patient's medical information in the EHR, there is no tool to support decisions made while planning the treatment and monitoring sessions of patients. The deployment of a decision support system to assist oncologists in therapy planning tasks is a real need that results in several benefits: workload of oncologists will be reduced and more time might be dedicated to personal assistance to patients (improving quality of health delivery), patient safety is augmented by automating administration rules, and efficiency of health delivery is increased since resource coordination and usage will be supported by an automated planning process that incorporates representation and reasoning about time and resources.

The following sections are devoted to describe how tasks concerning the stages of treatment and monitoring performed by oncologists, their internal process logic, and the temporal constraints to be observed during a treatment, can be represented by a temporally extended, HTN-based knowledge representation scheme. First, the main features of the HTN P&S system [6, 9], capable of managing such representation and used as the core technology to support oncologists' decisions on therapy planning will be summarized, then knowledge representation as well as planning and temporal reasoning aspects will be detailed.

3 MAIN FEATURES OF THE PLANNER

The AI Planning and Scheduling system used has been developed by our research group and, furthermore, has already been applied to other practical problems [9]. It uses as its planning domain and problem description language an HTN extension of PDDL (Planning Domain Description Language), a language used by most of well known planners that allows to represent non-hierarchical planning domains as a set of actions with typed parameters, preconditions and effects. Actions' effects are intended to represent changes in the world by defining which facts are asserted and retracted by the execution of actions. Numerical function are also allowed (what provides support to compute, for example, the duration of a drug-administration action depending on patient conditions) and, therefore, it is also possible to represent either consumable or discrete numerical resources (for example, the total drug dosage received by a patient, see *:durative-action* in Figure 1.(**b**)). Concretely, primitive tasks of our HTN−PDDL extension, are encoded as PDDL 2.2 level 3 durative actions (allowing to represent temporal information like duration and start/end temporal constraints, see [6] for details). In addition, HTN methods used to decompose compound tasks into sub-tasks include a precondition that must be satisfied by the current world state in order for the decomposition method to be applicable by the planner (see `(:task Protocol` in Figure 1.(**a**) that describes two alternative courses of action depending on the group a patient belongs to). The basic planning process is a state-based forward HTN planning algorithm that, starting from the initial state and a goal expressed as a high-level task, iteratively decomposes that top-level task and its sub-tasks by selecting their decomposition methods according to the current state and following the order constraints posed in tasks decomposition schemes as a search-control strategy.

This is a forward search process that makes the planner to know the current state of the world (internally represented as a set of facts that describe the context of the health-care treatment, including patient's current sate) at every step in the planning process. Concretely, this context-awareness is specially important when preconditions of both methods and primitive actions are evaluated, what allows to incorporate significant inferencing and reasoning power as well as the ability to infer new knowledge by requesting information to external hospital information services. In this sense, the planner uses two mechanisms addressed to represent as well as support oncologists decision-rules concerning issues like conserving patient safety on the administration of drugs. On the one hand, *deductive inference tasks* of the form `(:inline <p> <c>)` may be fired in the context of a decomposition scheme, when the logical expression `<p>` is satisfied by the current treatment state, providing additional bindings for variables or asserting/retracting literals into the planner's knowledge base, depending on the logical expression described in

`<c>`. These tasks can be used (as shown for example in Figure 1.(**b**)) to dynamically compute, depending on the current health-care context, the dosage an duration of drugs administration (from functions that define either the intensity of dosage or the time-rate, depending on the body surface of a patient). On the other hand, *abductive inference rules* of the form `(:derived <lit> <expr>)` allow to infer a fact `<lit>` by evaluating `<expr>`, that may be either a more complex logical expression or a Python script that both, binds its inputs with variables of `<lit>`, and returns information that might be bound to some of the variables of `<lit>`. For example, a derived literal might be used to infer whether a patient is in an correct state, from a complex expression including all the necessary conditions that enable the administration of a given drug (see derived literal on Figure 1.(**b**)). This literal might then be used as a precondition of an action that represents the task of administrating a drug.

3.1 Representing workflow patterns

Compound tasks, decomposition methods and primitive actions represented in a planning domain mainly encode the procedures, decisions and actions that oncologists must follow, according to a given oncology protocol, when they deal with a treatment on a given patient. More concretely, the knowledge representation language as well as the planner are also capable of representing and managing different workflow patterns present in any of such protocols (also present, on the other hand, in most CIGs formalisms [12, 15]). A knowledge engineer might then represent control structures that define both, the execution order (sequence, parallel, split or join), and the control flow logic of processes (conditional and iterative ones). For this purpose the planning language allows sub-tasks in a method to be either sequenced, and then they appear between parentheses (T1,T2) , or splitted, appearing between braces [T1,T2]. Furthermore, an appropriate combination of these syntactic forms may result in split, join or split-join control structs. For example, decomposition methods of the main task `Protocol` (Figure 1.(**a**)) describe that chemotherapy and radiotherapy sessions must be executed in parallel, but they must be synchronized with both a previous (split) and a later (join) evaluation of the general state of a patient (issues about temporal information included in the decomposition scheme shown will be detailed later).

Conditional and iterative control constructs can also be represented as task decomposition schemes that exploit the main search control techniques implemented by the planner. Briefly, a general process *p* that contains a conditional struct ***if*** *c* ***then*** *p1* ***else*** *p2* can be represented as a task decomposition scheme as the one shown in the task `Protocol` (Figure 1.(**a**)), that encodes a conditional structure based on the stratification group[4] of a patient. This decomposition scheme describes that if a condition *c* (a patient belongs to *Group1*) holds in the current health-care context, then apply `(:method Group1)` else apply `(:method Group2)`.

On the other hand, a general process *p* that contains an iterative struct ***while*** *c p1* may be represented as a task decomposition scheme as the one shown in the task `Chemotherapy` (Figure 1.(**b**)). This decomposition scheme describes that the primitive task `AdminDrug` should be repeatedly performed while the number of repetitions prescribed for the drug VCR (Vincristine) is greater than 0.

[4] Patients that receive a given protocol are initially stratified in a group depending on several criteria like the size of their tumour

3.2 Representing and reasoning about temporal constraints

Furthermore, our HTN domain description language as well as the planning algorithm support to explicitly represent and manage time and concurrency at every level of the task hierarchy in both compound and primitive tasks, by allowing to express temporal constraints on the start or the end of an activity. Any sub-activity (either task or action) has two special variables associated to it, `?start` and `?end`, that represent its start and end time points, and some constraints (basically `<=, =, >=`) may be posted on them (it is also possible to post constraints on the duration with the special variable `?duration`). In order to do that, any activity may be preceded by a logical expression that defines a temporal constraint as it is shown in `(:task Protocol` (Figure 1.(**a**)), where the duration of any chemotherapy session (an sub-tasks included in its decomposition) is constrained to 360 hours (15 days). The beginning of chemotherapy (in any of the two alternative courses of action) is constrained to start not earlier than a given date.

This temporal knowledge can be managed by the planning process thanks to the handling of metric time over a Simple Temporal Network (STN), a structure (X, D, C) such that X is the set of temporal points, D is the domain of every variable and C is the set of all the temporal constraints posted (See [6] for more details). In our case, a plan is deployed over a STN following a simple schema: every primitive action a_i included in a plan owns two time points $start(a_i)$ and $end(a_i)$, and every compound task t_i decomposed during the planning process generates two time points $start(t_i)$ and $end(t_i)$ which bound the time points of its sub-tasks. These temporal constraints are encoded as absolute constraints with respect to the absolute start point of a STN. All the time points share the same domain $[0, \infty)$, but it is important to note that the constraints in C (described in the planning domain) provide support to describe flexible temporal constraints, by defining earliest and latest execution times for start/end points associated to every task or action. For example, it is possible to encode constraints of the form `((and (>= ?start date1)(<= ?start date2)) (t))` what provides flexibility for the start time of `t`'s execution, indicating that `t` should start neither earlier than `date1` nor later than `date2`.

Every time that a compound or primitive task is added to the plan, all the time points and constraints of the STN are posted, propagated and validated automatically, observing both the implicit (derived from qualitative order constraints) and explicit (derived from quantitative constraints described in the domain) temporal constraints defined in any decomposition scheme. This temporal representation, on the one hand, provides enough expressivity power to truly represent workflow schemes such as sequence, parallel, split and join, since during the planning process our planner is capable of inferring quantitative temporal constrains from the qualitative ordering constraints expressed in decomposition methods. On the other hand, time points of subtasks of any task `t` with temporal constraints are embraced by the time points of `t`, what means that subtasks inherit the constraints of their higher-level task. This allows to represent and reason about temporal constraints derived from hierarchical decompositions, a strong requirement of any system devoted to support therapy planning (as stated in [17]).

The process and representation so far described present some advantages with respect to current state of the art techniques devoted to therapy planning that are worth to note. Firstly, the representation and reasoning about temporal constraints of our approach allows to simultaneously validate temporal constraints while generating therapy plans (plan generation and temporal constraint management are interleaved). Most approaches [3] are only focused in one side of the problem of therapy planning, since they only pay attention on how to manage temporal constraints of actions, and neglect aspects related to how automatically generate sequences of actions with temporal constraints. Very few [8, 18] face the problem of plan generation, but it is carried out following a static, non-deliberative process (close to case-based planning), that is not interleaved with temporal constraints reasoning. Instead of this, it is based on a batch process that firstly generates a complete plan and then analyzes its temporal constraints, what affects negatively to the efficiency of the overall process, as well as to important reasoning aspects like the loss of backtracking points (which are lost when a plan is completely generated) or the impossibility of using the causal rationale of the plan as a guide to propagate constraints (as is the case of our planner [6]). These features are specially important when plans have to be readapted due to new circumstances arisen during the treatment stage.

3.3 Representing periodic tasks and temporal constraints

The HTN planner is also able to record the start and end of any activity and to recover these records in order to define complex synchronization schemes between either tasks or actions as relative constraints with respect to other activities. This mechanism is used to encode synchronization of tasks that correspond to repetitive periodic patterns. The first step is the definition, by assertion, of *temporal landmarks* that signal the start and the end of either a task or an action (Figure 2). These landmarks are treated as PDDL fluents (predicates that represent functions which when evaluated return a value or an object, in this case, a timepoint of the STN) that are associated to the time points of the temporal constraints network.

```
(:durative-action AdminDrug
 :parameters (?p - Patient ?ph - Drug ?ds ?dur - number)
 :duration (= ?duration ?dur)
 :condition (patient_ok ?p)
 :effect (and (increase (total_dosis ?p ?ph) ?ds))
              (assign (last-admin ?p ?ph) ?end))

(:task A3
 :parameters (?p - Patient ?ph - Drug)
 (:method A3
  :precondition (...)
  :tasks (((= ?start (last-admin ?p ?ph)) (b)))))
```

Figure 2. Generating and recovering a temporal landmark.

These landmarks are asserted in the planner's current state, and later on, they may be recovered and posted as constraints of other tasks in order to synchronize two or more activities. For example, Figure 2 shows how to recover a temporal landmark that restricts action `b` to start exactly at the same time than action `AdminDrug` ends.

In particular, thanks to the expressive power of temporal constraints networks and to the mechanism explained so far, a planning domain designer may explicitly encode in a problem's domain all of the different orderings included in Allen's algebra [2] between two or more tasks, between two or more actions or between tasks and actions. Furthermore, temporal landmarks are an excellent resource in order to express different kinds of periodic patterns to be followed by temporal constraints, a strong requirement of clinical protocols, particularly oncology clinical protocols. For example, Figure 3 shows a refined description of the `Chemotherapy` task that combines temporal landmarks management and recursive decompositions in order to specify that the administration of VCR must be always preceded by a delay of 24 hours, and must be repeated a number of times defined by a function (`(NRep ?p VCR)`). Additionally, note that all the actions of this chemotherapy cycle must be executed in an interval of 15 day (360 hours), since the task `Chemotherapy` has been constrained to a duration of 360 hours (15 days), as shown in Figure 1.(**b**), and the planning process allows subtasks to inherit constraints of higher-level tasks.

```
(:task ChemoTherapy
 :parameters (?p - Patient)

 (:method repeat
  :precondition (> (NRep ?p VCR) 0)
  :tasks (
   (:inline () (decrease (NRep ?p VCR)))
   (:inline () (assign ?dosage (* (surface ?p) (intensity ?p))))
   (:inline () (assign ?dur (* (surface ?p) (time_rate ?p))))
   ((and (>= ?start (last-admin ?p VCR)) (= ?duration 24))
    (Delay ?p VCR))
   ((and (= ?duration ?dur))
    (AdminPharmac ?p VCR ?dosage ?dur))))

 (:method base_case
  :precondition (= (NRep ?p VCR) 0)
  :tasks ()))
```

Figure 3. A chemotherapy cycle

4 PROOF OF CONCEPT

Considering the previous description, a proof of concept of this technology has been carried out in collaboration with expert oncologists in the Hospital Complex of Jaén (Spain). During this proof, a model of a concrete oncology clinical trial protocol (the one followed at present to planning the treatment of Hodgkin's disease [11] and elaborated by the Spanish Society on Pediatrics Oncology) has been encoded in the planning language above described, in a knowledge elicitation process based on interviews with experts. In the experiments performed, the planner received the following inputs: a planning domain, representing this protocol; an initial state representing some basic information to describe a patient profile (stratification group, age, sex, body surface, etc.) as well as other information needed to apply administration rules about drugs (dosage, frequency, etc.); and a high-level task representing the goal (apply the protocol to the patient) with temporal constraints representing the start date of the treatment plan. The output of the planner are plans with actions temporally annotated with start/end constraints. These plans are represented in a standard XML representation and may be visualized as Gantt charts in standard tools devoted to project management (like MS Project, see Figure 4). Several experiments were realized on different patient profiles, and all the plans were obtained in less than one second. The domain includes six compound tasks, 13 methods, 6 primitive tasks and the file contains more than 400 lines of code [5]. Plans generated represent therapy plans tailored to a given patient profile, and they allow to represent therapies of more than one year of duration, including more than 50 actions.

Plans contain actions that represent activities as well as decisions an oncologist should follow, and they are deployed over a STN used to represent time intervals that constraint both start and end execution times of actions. Therefore, at the beginning of the execution

[5] Available on `http://decsai.ugr.es/~faro/Hodgkin/index.html`

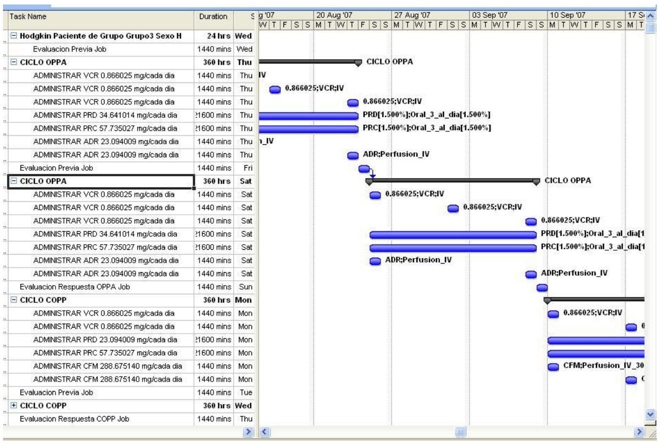

Figure 4. A temporally annotated and automatically generated therapy plan represented as a gantt chart

of a therapy plan, actions, temporal constraints and facts that represent preconditions and effects of actions are consistent with respect to the initial conditions expressed in the planning problem. Additionally, regarding plan execution, a monitoring process has been developed (applied to a different domain application [9] that, nevertheless, shares this same plans representation) that guarantees the correct execution of actions, thus avoiding for example the activation of actions once they have been finished. However, as the plan execution is progressing, inconsistencies may arise that could affect either the temporal dimension of the plan or actions' preconditions. In such cases a rescheduling process might be carried out devoted to rearrange temporal constraints, by checking the consistency of the underlying plan's temporal network. In the case that a consistent temporal network couldn't be found, an automated replanning process (based on the same planning process here described) might be triggered in order to readapt the therapy plan to new circumstances.

5 CONCLUSIONS

In this work we have presented an AI P&S system based on temporal Hierarchical Task Networks (HTN) planning techniques able to automatically and dynamically generate personalized therapy plans for oncology patients, following a deliberative hierarchical planning process driven by the procedural knowledge described in oncology protocols. This approach should not be considered only as a new way to represent therapies. Regarding other approaches devoted to therapy plan management (like Asbru [8] or Glare [17]), authors argue that therapy planning is not supported in these systems by an automated, deliberative process as the one presented in this work. Instead, the plan management life-cycle of these approaches requires specialized human intervention (either knowledge engineers or trained medical staff) when tailoring a therapy plan from an initial protocol scheme to a given patient profile. These approaches are mainly focused on the verification of therapy plans with temporal constraints (apart from providing very expressive CIGs representation formalisms) and we have shown that our temporal representation and reasoning is as expressive as the one used in Asbru or Glare. Furthermore, the process performed by these approaches to temporal constraints verification could be used at execution time in order to revise possible temporal inconsistencies (like a delay in the administration of a drug), but there are circumstances in which the actions included in a therapy plan (and not only temporal constraints) must be partial or completely readapted (for example, when a patient's stratification group changes since his/her tumour size does not progress as expected). In such cases our approach might to use the same planning process to automatically readapt the therapy plan, leveraging the whole life cycle of the treatment, by shifting more detailed decisions to the planner and reducing the workload of oncologists, as opposite to current approaches that always need to readapt from the scratch.

Finally, we cannot neglect the use of standard languages and frameworks for modeling and editing CIGs. Indeed, our next planned step is to represent oncology clinical protocols into one of these standard schemes and to develop a fully automated translation process from such representation to our planning language, thus allowing to automatically generate, execute and monitor treatment plans from a standard representation.

REFERENCES

[1] Syed Sibte Raza Abidi, 'Knowledge management in healthcare: towards 'knowledge-driven' decision-support services', *International Journal of Medical Informatics*, **63**, 5–18, (2001).

[2] J.F. Allen, 'Maintaining knowledge about temporal intervals', *Comm. ACM*, **26**(1), 832–843, (1983).

[3] J.C. Augusto, 'Temporal reasoning for decision support in medicine', *Artificial Intelligence in Medicine*, **33**, 1–24, (2005).

[4] K. Bradbrook, G. Winstanley, D. Glasspool, J. Fox, and R. Griffiths, 'Ai planning technology as a component of computerised clinical practice guidelines.', *Lecture Notes in Artificial Intelligence*, **3581**, 171–180, (2005).

[5] J. L. Bresina, A. K. Jonsson, P. Morris, and K. Rajan, 'Activity planning for the mars exploration rovers', in *Proceedings of the ICAPS05*, pp. 40–49, (2005).

[6] L. Castillo, J. Fdez-Olivares, O. García-Pérez, and F. Palao, 'Efficiently handling temporal knowledge in an HTN planner', in *Proc. ICAPS*, (2006).

[7] E. Coiera, *Guide to Health Informatics*, chapter Clinical Decision Support Systems, 2003.

[8] G. Duftschmid, S. Miksch, and W. Gall, 'Verification of temporal scheduling constraints in clinical practice guidelines', *Artificial Intelligence in Medicine*, (2002).

[9] J. Fdez-Olivares, L. Castillo, O. García-Pérez, and F. Palao, 'Bringing users and planning technology together. Experiences in SIADEX', in *Proc. ICAPS*, (2006).

[10] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning. Theory and Practice*, Morgan Kaufmann, 2006.

[11] SEOP Working Group, 'National protocol for diagnosis and treatment of hodgkin's desease (in spanish).eh-seop.003', Technical report, SEOP, (2003).

[12] T.Y. Leong, K. Kaiser, and S. Miksch, 'Free and open source enabling technologies for patient-centric, guideline-based clinical decision support: A survey.', *Methods of Information in Medicine*, **46**, 74–86, (2007).

[13] Nataliya Mulyar, Wil M.P. van der Aalst, and Mor Peleg, 'A pattern-based analysis of clinical computer-interpretable guideline modeling languages', *JAMIA*, (2007).

[14] D. Nau, H. Muoz-Avila, Y. Cao, A. Lotem, , and S. Mitchel, 'Total-order planning with partially ordered subtask', in *Proceedings of the IJCAI-2001*, (2001).

[15] M. Peleg, SW. Tu, J. Bury, P. Ciccarese, J. Fox, and RA. Greenes RA, 'Comparing computer-interpretable guideline models: A case-study approach', *J Am Med Inform Assoc*, **10**, 10–58, (2003).

[16] C. Spyropoulos, 'Ai planning and scheduling in the medical hospital environment', *Artificial Intelligence in Medicine*, **20**(101-111), (2000).

[17] P. Terenziani, L. Anselma, S. Montani, and A. Bottrighi, 'Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines', *Artificial Intelligence in Medicine*, **38**(2), 171–195, (2006).

[18] P. Votruba, A. Seyfang, M. Paesold, and S. Miksch, 'Environment-driven skeletal plan execution for the medical domain', in *European Conference on Artificial Intelligence (ECAI-2006)*, pp. 847–848.