Computational Intelligence
An International Journal

Computational Intelligence, Volume 27, Number 1, 2011

SUPPORTING CLINICAL PROCESSES AND DECISIONS BY HIERARCHICAL PLANNING AND SCHEDULING

JUAN FDEZ-OLIVARES,[1] LUIS CASTILLO,[1] JUAN A. CÓZAR,[2] AND OSCAR GARCÍA PÉREZ[3]

[1]*Department of Computer Science and A.I., University of Granada, Granada, Spain*
[2]*Pediatrics Oncology Service, Hospitalary Complex of Jaén, Jaén, Spain*
[3]*R&D Department, Iactive Intelligent Solutions, S. L., Granada, Spain*

This article is focused on how a general-purpose hierarchical planning representation, based on the hierarchical task networks (HTN) paradigm, can be used to support the representation of oncology treatment protocols. The planning algorithm used is a temporally extended HTN planning process capable of interpreting such representation and generating oncology treatment plans that have been proven to support clinical decisions in the area of pediatrics oncology.

Key words: AI planning and scheduling, temporal hierarchical task networks planning, clinical decision support systems.

1. INTRODUCTION

Hierarchical planning, and more concretely hierarchical task networks (HTN) planning (Sacerdoti 1975; Tate 1977; Castillo et al. 2006), is a planning paradigm that supports the modeling of planning domains in terms of a compositional hierarchy of tasks representing compound and primitive tasks at different levels of abstraction. A hierarchical planning algorithm mainly decomposes compound tasks into (compound/primitive) subtasks, following the order constraints described in different (and possible alternative) decomposition methods. It follows a reasoning process driven by the procedural knowledge encoded in the HTN domain, to determine how to perform a high-level task introduced as problem. This planning paradigm, from a practical point of view, can not only be seen (as is classified in Ghallab, Nau, and Traverso 2006) as another way to represent heuristic and control knowledge to speed up planners, by introducing ad hoc procedural knowledge that guides the search of a primitive action-based planner. Indeed, the knowledge representation scheme on which HTN planning is based is a necessary way to face a great part of practical problems (Bresina et al. 2005; Castillo et al. 2007). This work is particularly focused on those in which humans need either to solve problems or carry out their work or making decisions guided by the know-how of a given organization described in preexisting operating procedures or protocols. In such cases, the main criticism received by this planning paradigm (the additional knowledge representation effort for an HTN planner to work that can be eluded by other means) becomes a need. This is the case of the medical domain, and more concretely the field of therapy planning systems (Spyropoulos 2000; Augusto 2005; Votruba et al. 2006), which are aimed to recommend predefined general courses of action to be applied to a patient, on the process of treating a disease.

Therapy planning systems incorporate, on the one hand, a computerized representation of clinical protocols, also called *Computer Interpretable Clinical Guidelines* (CIGs) (Peleg et al. 2003). A clinical protocol describes evidence-based operating procedures that physicians follow as a guide to perform clinical tasks as well as making clinical decisions. Most of the research and development effort on these systems has been focused on the development of languages and frameworks to support modeling, editing, and representing CIGs (Leong,

Address correspondence to Juan Fernández Olivares, Departamento de Ciencias de la Computación e I.A., ETSI Informática y Telecomunicación, Universidad de Granada, 18071, Granada, Spain; e-mail: faro@decsai.ugr.es

Kaiser, and Miksch, 2007). Furthermore, all of them are based on *Task Networks Models* (Peleg et al. 2003) where mechanisms to represent workflow patterns (Mulyar, van der Aalst, and Peleg 2007) that describe the process logic between subtasks are also included (mainly sequential, conditional, iterative, and synchronization control structures). On the other hand, although less effort has been devoted to develop techniques to operationalize such representations, some systems (Augusto 2005; Duftschmid, Miksch, and Gall 2002; Terenziani et al. 2006) 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.

In principle, it seems that HTN planning is an appropriate technique that might support both the representation of clinical processes and clinical decision making in therapy planning. An HTN planning process might take advantage of its deliberative and knowledge-driven reasoning process to automatically generate treatment plans, starting from an accurate representation of clinical protocols. However, to authors' knowledge, there is no application of HTN techniques to this field. It might be due to the fact that the great part of these approaches have centered on temporal constraints reasoning (Duftschmid et al. 2002; Terenziani et al. 2006) aimed to validate constraints on a previously generated, hand-tailored treatment plan (Votruba et al. 2006), but very little attention has been paid to the automated generation of therapy plans (Spyropoulos 2000; Bradbrook et al. 2005). In addition, an argument used to reject the application of these techniques (Augusto 2005) is the lack of support for a flexible execution of plans obtained.

Considering that the management of time is a crucial requirement to be fulfilled by any application to therapy planning, and trying to demonstrate the usefulness of HTN techniques in the medical domain, in this article 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 into the reasoning process to obtain temporally valid plans, suitable to be applied and flexibly executed as oncology treatment 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 pediatrics oncology area, in which health assistance (and particularly therapy planning) is based on the application of oncology treatment protocols: a set of evidence-based operating procedures and policies to be followed in both stages of the care providing life cycle, treatment planning, and treatment 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 (Mulyar et al. 2007), 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. Evaluation and follow-up 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 either their relative order or deadline goals and 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 (Terenziani et al. 2006).

In addition, treatment sessions established by any protocol must be arranged considering the availability and capacity of human and material resources (this is not a matter of an oncology protocol but it is necessary to put it in practice). Working shifts of oncologists must be taken into account when planning evaluation and follow-up sessions as well as capacity and availability dates for administration beds or hospital test facilities needed to obtain clinical tests to study the evolution of the patient. 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 (pediatrics oncology services in the public health system of Andalusia) is done by hand, that is, though it is possible to access patient's medical information in electronic health records (EHRs), 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 because resource coordination and usage will be supported by an automated planning process capable of representing 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 and resource 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 (Castillo et al. 2006; Fdez-Olivares et al. 2006), 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 (Castillo et al. 2006) and, furthermore, it has already been applied to other practical problems (Fdez-Olivares et al. 2006). It uses as its planning domain and problem description language an HTN extension of Planning Domain Description Language (PDDL), a language used by most of well-known *primitive action-based* planners that allows to represent nonhierarchical planning domains as a set of actions with typed parameters, preconditions, and effects. The effects of actions are intended to represent changes in the world by defining which facts are asserted and retracted by the execution of actions. Numerical functions are also allowed (what provides support to compute, e.g., the duration of a drug-administration action depending on patient conditions) and, therefore, it is also possible to represent discrete numerical resources (e.g., 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, such as, duration and start/end temporal constraints; see Castillo et al. 2006 for details). In addition, HTN methods

(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)
   (ChemoTherapy ?p)))

 (: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*.

- Set $\mathcal{A}$, the agenda of remaining tasks to be done, to the set of high level tasks speci ed in the goal.
- Set $\Pi = \emptyset$, the plan.
- Set $\mathcal{S}$, the current state of the problem, to be the set of literals in the initial state.

1. Repeat while $\mathcal{A} \neq \emptyset$
   (a) **Extract** a task t from $\mathcal{A}$
   (b) if t is a primitive action, then
      i. If $\mathcal{S}$ satis es t preconditions then
         A. Apply t to the state, $\mathcal{S} = \mathcal{S} + additions(t) - deletions(t)$
         B. Insert t in the plan, $\Pi = \Pi + t$
         C. **Propagate-Temporal-Constraints**(Π)
      ii. Else **FAIL**
   (c) if t is a compound action, then
      i. If there is no more decomposition methods for t then **FAIL**
      ii. **Choose** one of its decomposition methods of t whose preconditions are true in $\mathcal{S}$ and map t into its set of subtasks $t_1, t_2, \ldots$
      iii. Insert $t_1, t_2, \ldots$ in $\mathcal{A}$.
2. **SUCCESS**: the plan is stored in Π.

FIGURE 2. A rough outline of the HTN planning algorithm showing the point at which temporal constraints are propagated (step *1.(b).i.C*).

used to decompose compound tasks into subtasks 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 (shown in Figure 2) receives as input a set of facts that represent an initial state of the world (that describes the context of the healthcare treatment, including patient's current state) as well as a goal, described as a partially ordered set of tasks that need to be carried out. Then it follows a state-based forward HTN planning algorithm

that decomposes that top-level set of tasks and its subtasks 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. Therefore, it follows a *deliberative* reasoning process that explores the space of possible decompositions replacing a given task by its component activities (that may be either primitive or compound), until the initial set of tasks is transformed into a set of only primitive actions that make up the plan. A key feature of this planning system is that the deliberative planning process is interleaved with a temporal reasoning process supported by a simple temporal constraints network that underlies the plan in construction. More detail about the temporal representation and reasoning will be given later.

The forward search and decomposition process makes the planner to know, at every step in the planning process, the current context of the healthcare treatment, including patient's current state. Concretely, this context awareness is specially important when preconditions of both primitive actions and methods are evaluated at planning time on the current state (respectively, steps (b)i and (c)ii in Figure 2), which 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 oncologists' decision rules concerning issues, such as, 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, e.g., in Figure 1(b)) to dynamically compute, depending on the current healthcare context, the dosage of a 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>` in the current state , which 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 is 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; Peleg et al. 2003; Leong et al. 2007). 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 subtasks 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 node) and a later (join node) 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[1] of a patient. This decomposition scheme describes that if a condition *c* (a patient belongs to *Group1*) holds in the current healthcare 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 zero.

3.2. Representing and Reasoning about Temporal Constraints

Furthermore, our HTN domain description language as well as the planning algorithm supports 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 subactivity (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`). 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 (subtasks 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 Castillo et al. 2006, 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 subtasks. 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)`

[1] Patients who receive a given protocol are initially stratified in a group depending on several criteria, such as, the size of their tumor.

```
(: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 3. Generating and recovering a temporal landmark.

(<= ?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 because 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, which 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 Terenziani et al. 2006).

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 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 3). 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.

These landmarks are asserted in the planner's current state, and later on, they may be recovered and posted as constraints of other tasks to synchronize two or more activities. For example, Figure 3 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

```
(: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))(AdminDrug ?p VCR ?dosage ?dur))
   (Chemotherapy ?p)))

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

FIGURE 4. A chemotherapy cycle.

domain all of the different orderings included in Allen's algebra (see Castillo et al. 2006, for details) between two or more tasks, between two or more actions or between tasks and actions. Furthermore, temporal landmarks are an excellent resource 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 4 shows a refined description of the `Chemotherapy` task that combines temporal landmarks management and recursive decompositions 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 days (360 hours) because 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.

3.4. Representing and Managing Resources

The workflow specified in an oncology treatment protocol does not include issues related to which human and material resources are involved in the therapy planning process, but it is necessary to represent and manage them to truly support clinical processes and decisions. Therefore, capacity and availability dates of consumable, discrete resources may be represented in the planning domain description language. A generalization of timed initial literals (Castillo et al. 2006), that allows to represent temporal patterns for exogenous events, is used to this end.

For example, as shown in Figure 5, the `(between ...)` clause (specified in the planning problem) represents periods of 24 hours of availability of an oncologist, repeated every week. Thus, evaluation sessions that require the presence of a specialist, must be scheduled only when the oncologist is available. This is modeled as a `(at start...)` precondition in the proper primitive action `eval-patient`. The dates in which the literal is true are represented as time points and, because this literal may appear several times with different associated time points, it also represents a choice point and, therefore, a backtracking point for the satisfaction of the precondition of action `eval-patient`.

```
(between "07/08/2007 00:00:00" and "08/08/2007 00:00:00"
         and every 144hrs (available John))
....
(:durative-action eval-patient
 :parameters (?p - Patient ?s - specialist)
 :duration (= ?duration 24hrs)
 :condition (and (at start (available ?s)) ...)
  .....
```

FIGURE 5. Oncologists' working shifts and how this information is used as preconditions in evaluation sessions.

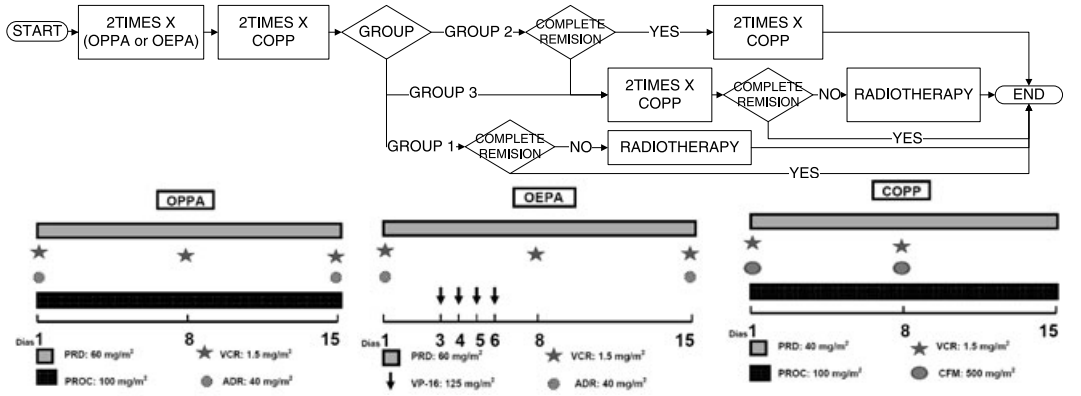

FIGURE 6. A general schema of Hodgkin's disease clinical protocol. The representation followed to show the periodical temporal patterns for chemotherapy cycles (OPPA, OEPA, and COPP) is literally copied from the protocol specification. OPPA, OEPA and COPP are protocol-internal identifiers used to denote three qualitatively different cycles of chemotherapy.

It is necessary to note that the search and reasoning process that supports the planning algorithm of the planner is not intended to obtain an optimal assignment of resource constraints, instead of this, the planning and scheduling process obtains the first feasible plan with a correct arrangement of actions and temporal and resource constraints.

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 treatment protocol (the one followed at present for planning the treatment of Hodgkin's disease (Group 2003) 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.

A general schema of the treatment workflow process, indicated in such clinical protocol, is outlined in the flow-chart diagram of Figure 6. First, a child must receive two chemotherapy cycles (of type OPPA or OEPA, depending on the genre) and another two cycles of type COPP. If a complete remission of the tumor is not achieved by patients of *Group1* then radiotherapy

sessions must start. If the stratification group (decided by the oncologist) is either *Group2 or Group3* two more COPP cycles must be administrated. In case of a patient of *Group3*, additional radiotherapy sessions must be administrated when a complete remission of the tumor is not detected.

Temporal patterns to administrate every type of chemotherapy cycle are shown below the flow chart of Figure 6. For example, a cycle of type OPPA takes 15 days, the rules to administrate a cycle of type OPPA state that the drugs prednisone and procarbazine must be administrated every day (dosage is also specified), VCR has to be administrated the first, eighth, and last day, and ADR the first and last day (OEPA and COPP patterns should be interpreted in a similar way). In addition, start times for every chemotherapy cycle must be separated at least 28 days, and an evaluation session has to be scheduled previously to the start of every cycle.

As previously explained, the decision-making process followed by clinicians during the elaboration of a treatment plan must be guided by the knowledge specified in the clinical treatment protocol that, as shown in this example, embodies knowledge about clinical processes and about clinical decisions. This decision-making process results in a complete treatment plan composed by a sequence of temporally annotated clinical actions, with a time horizon that covers all the treatment life cycles: chemotherapy, evaluation, and radiotherapy sessions. It is important to remark that the process represented in a treatment protocol might be seen as a problem-solving strategy aimed to achieve a complete remission of a given tumor: it establishes multiple ways to refine high-level tasks into lower level ones and the available options are constrained mainly by clinical data about a patient. These multiple courses of actions might also be interpreted as alternative refinements that reflect different strategies and tactics available to achieve a complete remission of a tumor. This problem-solving strategy clearly fits to the knowledge-driven, deliberative reasoning process followed by the planning technology described in previous sections. Therefore, workflow patterns included in the treatment protocol, temporal constraints to be observed between chemotherapy cycles, periodic patterns to administrate drugs as well as the representation of oncologists' working shifts have been encoded as planning domain and problem files. The domain includes 6 compound tasks, 13 methods, 6 primitive tasks and the file contains more than 400 lines of code.[2] A discussion about experiments performed and plans obtained is shown in the next section.

5. EXPERIMENTAL RESULTS

In every experiment performed, the planner received as input a planning domain, representing the Hodgkin's disease protocol (in the terms described in the previous section), and a therapy planning problem in the following terms:

- An initial state representing some basic information to describe a patient profile (stratification group, sex, body surface, etc.).
- Temporal constraints to be considered in chemotherapy cycles as well as in the administration of drugs (drug dosage, frequency and administration mode).
- Resource constraints, mainly related to availability of resources, taking the form of working shifts of oncologists.
- A task-goal at the highest level of abstraction that is interpreted by the planner as a high-level specification of the strategy to be followed to generate a therapy plan. This

[2] Available on http:://decsai.ugr.es/∼faro/Hodgkin/index.html

task goal can be annotated with temporal constraints (representing the start date of the treatment plan) and it is subsequently decomposed by the planner following the strategy declared in the HTN domain.

Figure 7 shows an example problem file where some basic information about date format (dd/mm/yy), objects managed by the planner, and initial data for a given patient are represented. Resource availability constraints are also represented as working shifts of two oncologists in such a way that each one of them is alternatively available to attend evaluation sessions during 1 month. In addition to this information represented by a PDDL extension of timed literals, temporal constraints on chemotherapy cycles and drug-administration rules are also declared taking into account the following properties:

- Dosage (mmg/cm^2) and administration mode: `IV` for intravenous, `IVP` for intravenous perfusion and `O3D` for oral administration three times at day.
- Frequency of administration: the PDDL function `(NRep ?drug ?cycle)` represents the number of times a drug must be administrated in a cycle, that is, the number of repetitions of an action in a cycle.
- Duration of administration: the literal `(rep_dur ?drug ?cycle ?dur)` represents the duration of a single drug-administration action (i.e., a repetition) in a cycle.
- Elapsed time between repetitions: the time between every repetition in a cycle is represented by the literal `(inBetweenAll ?drug ?cycle ?dur)`. This literal is useful when the elapsed time corresponds to a periodic pattern. It is also possible to encode ad hoc elapsed durations with the literal `(inBetweenEach ?i ?drug ?cycle ?dur)` that defines the elapsed duration between the consecutive repetitions $i - 1$ and i. This literal has not been used in these experiments because the different chemotherapy cycles do not require it.

For example, as shown in the problem of Figure 7, the drug VCR in an OPPA cycle must be administrated three times, each one with a duration of 24 hours (that is, it has to be scheduled so that it has to be administrated at a given day) and considering an elapsed time of 6 days (144 hours) between all the administration actions (in other words, VCR must be administrated the days 1st, 8th, and 15th of the OPPA cycle, as shown previously in Figure 6). This information represented in the problem is managed at planning time as temporal constraints. In addition, other knowledge related to total cycle duration, cycle's start and end dates is declared in the HTN domain as specific temporal constraints of tasks, at different levels of abstraction, as explained in previous sections.

Considering this problem description, the planning problem consists of finding a sequence of temporally annotated clinical actions that fits to the guidelines declared in the clinical protocol (and represented in the HTN domain), subject to the resource availability constraints described in this case as working shifts of oncologists, and to the temporal constraints declared as temporal patterns for chemotherapy administration. Several experiments have been performed on six patients, each one considered as a representative of the six different profiles corresponding to the possible combinations of sex and stratification group: female (Group1, Group2, Group3) and male (Group1, Group2, Group3). Experiments were aimed to validate both the knowledge represented (Hodgkin's protocol) and therapy plans generated. Oncologists played a validation role (based on interviews) in both stages, knowledge elicitation and representation, and plan validation. Plans generated represent therapy plans tailored to a given patient profile, and they allow to represent therapies of several months of duration, including tens of temporally annotated actions with start/end constraints. Figure 8 shows, for each patient profile, the total number of actions, the number of chemotherapy

```
(define (problem hodgkin1) (:domain hodgkin)(:customization
(= :time-format "%d/%m/%Y")
(= :time-horizon-relative 2500)
(= :time-start "08/11/2007 08:00:00")
(= :time-unit :hours))


(:objects
 Job - Patient
 Juan Tomas - Specialist
 PRD VCR PRC ADR VP16 CFM - Drug
 IV O3D IVP - Mode
 Group1 Group2 Group3 - Group
 M F - Sex

)
(:init

;;Patient profile
    (= (body-surface Job) 10)
    (= (weight Job) 15.0)
    (= (height Job) 80.0)
    (sex Job F)
    (group Job Group3)
;;Patient current state
    (= (leucocites Job) 2500)
    (= (neutrophiles Job) 600)
    (= (plaquets Job) 110000)

;; Working Shifts
 (between "07/11/2007 00:00:00" and "07/12/2007 00:00:00" (available Juan))
 (between "07/12/2007 00:00:00" and "07/01/2008 00:00:00" (available Tomas))
 (between "07/01/2008 00:00:00" and "07/02/2008 00:00:00" (available Juan))
 (between "07/02/2008 00:00:00" and "07/03/2008 00:00:00" (available Tomas))
....

(start-date Job "08/11/2007 08:00:00")

;;VCR Administration in cycle OPPA
(= (dosage VCR OPPA) 1.5)
(mode VCR IV)
(= (NRep VCR OPPA) 3)
(rep_dur VCR OPPA 24)
(inBetweenAll VCR OPPA 144)

;;PRD Administration in cycle OPPA
(= (dosage PRD OPPA) 60.0)
(mode PRD O3D)
(= (NRep PRD OPPA) 15)
(rep_dur PRD OPPA 24)
(inBetweenAll PRD OPPA 0)

...
)
(:tasks-goal
   :tasks( (and (>= ?start "08/11/2007 08:00:00")) (Hodgkin Job)))
)
```

FIGURE 7. A therapy planning problem using an extended PDDL syntax to encode timed literals.

Sex	Group	Actions	Cycles	Duration	Time
Female	Group1	35	4	4	0.26
Female	Group2	51	6	5	0.43
Female	Group3	51	6	5	0.44
Male	Group1	35	4	4	0.13
Male	Group2	51	6	5	0.44
Male	Group3	51	6	5	0.43

FIGURE 8. Number of therapy actions, number of chemotherapy cycles, duration of therapy (in months) and time performance (in seconds) of therapy plan generation for different patient profiles (on an Intel Core2 Duo 2.40 GHz CPU).

cycles, the total therapy duration (in months), and the generation time (in seconds) of its corresponding therapy plan.

Figure 9 shows one of these therapy plans (with 51 actions and 6 months of duration) obtained for a female patient of stratification group *Group3*. For each action in the plan, the following information is shown: start and end dates, duration (in hours), task name, and resources employed. In the case of drug-administration actions, the resource is a drug, and the plan also informs about its dosage and administration mode. Evaluation sessions have an oncologist as resource, and it is used depending on his/her working shift. In addition, the plan shows summary tasks that inform about the start, end, and duration of chemotherapy cycles. These tasks are generated at planning time and they are included for informative purposes, helping to structure the final therapy plan.

The protocol also states different types of evaluation sessions that are shown in the therapy plan of Figure 9: a previous evaluation must be scheduled before every chemotherapy cycle (if no other type of evaluation is scheduled), a response evaluation must be scheduled after the administration of two chemotherapy cycles of the same type, and a tumor remission evaluation must be scheduled at the end of the treatment. It is important to note that evaluation sessions are scheduled according to both working shifts of oncologists and specific temporal constraints between chemotherapy cycles (recall that chemotherapy cycles must be administrated with a separation of 28 days).

An important issue that raised when plans were shown to oncologists during plan validation is related to the detail of a therapy plan. For example, the administration of drugs with oral mode might be subdivided into more detailed repetitions (one for each administration and scheduled at different hours each day). However, this detail was not considered informative for oncologists. The same happens to the actions that have to be applied continuously (every day) during a chemotherapy cycle, as the action to administrate either PRD or procarbazine (see Figure 9). In this case, oncologists recommended to show them as a single, summarized action, instead of a set of repetitions, because it is better suited by third party visualization tools (as explained below).

These plans are represented in a standard Extensible Markup Language (XML) representation and may be displayed as Gantt charts in standard tools devoted to project management (such as, MS Project, see Figure 10). The plan shown in Figure 10 has been obtained after an automated postprocessing of the output of the planner, to friendly show the tasks of the plan (left-hand side of the figure) as well as their temporal dimension as a Gantt chart (right-hand side). The visualization of the tasks in a MS Project display allows to show tasks in a *Work Breakdown Structure* including different outline levels (either *summary tasks* as OPPA CYCLE or *standard tasks* as AdminDrug), which may be collapsed or deployed as

HODGKIN'S PROTOCOL SEX FEMALE GROUP GROUP3

Start	End	Dur (hrs.)	Task	Resource	Mode	Dosage (mmg)
07/12/2007	08/12/2007	24	Previous Eval.	Tomas		
08/12/2007	23/12/2007	360	Summary OPPA			
08/12/2007	09/12/2007	24	AdminDrug	VCR	IV	0.866025
15/12/2007	16/12/2007	24	AdminDrug	VCR	IV	0.866025
22/12/2007	23/12/2007	24	AdminDrug	VCR	IV	0.866025
08/12/2007	23/12/2007	360	AdminDrug	PRD	O3D	34.641014
08/12/2007	23/12/2007	360	AdminDrug	PRC	O3D	57.735027
08/12/2007	09/12/2007	24	AdminDrug	ADR	IVP	23.094009
22/12/2007	23/12/2007	24	AdminDrug	ADR	IVP	23.094009
07/01/2008	08/01/2008	24	Previous Eval.	Juan		
08/01/2008	23/01/2008	360	Summary OPPA			
08/01/2008	09/01/2008	24	AdminDrug	VCR	IV	0.866025
15/01/2008	16/01/2008	24	AdminDrug	VCR	IV	0.866025
22/01/2008	23/01/2008	24	AdminDrug	VCR	IV	0.866025
08/01/2008	23/01/2008	360	AdminDrug	PRD	O3D	34.641014
08/01/2008	23/01/2008	360	AdminDrug	PRC	O3D	57.735027
08/01/2008	09/01/2008	24	AdminDrug	ADR	IVP	23.094009
22/01/2008	23/01/2008	24	AdminDrug	ADR	IVP	23.094009
23/01/2008	24/01/2008	24	Response Eval. OPPA	Juan		
05/02/2008	20/02/2008	360	Summary COPP			
05/02/2008	06/02/2008	24	AdminDrug	VCR	IV	0.866025
12/02/2008	13/02/2008	24	AdminDrug	VCR	IV	0.866025
05/02/2008	20/02/2008	360	AdminDrug	PRD	O3D	23.094009
05/02/2008	20/02/2008	360	AdminDrug	PRC	O3D	57.735027
05/02/2008	06/02/2008	24	AdminDrug	CFM	IVP	288.675140
12/02/2008	13/02/2008	24	AdminDrug	CFM	IVP	288.675140
20/02/2008	21/02/2008	24	Previous Eval.	Tomas		
04/03/2008	19/03/2008	360	Summary COPP			
04/03/2008	05/03/2008	24	AdminDrug	VCR	IV	0.866025
11/03/2008	12/03/2008	24	AdminDrug	VCR	IV	0.866025
04/03/2008	19/03/2008	360	AdminDrug	PRD	O3D	23.094009
04/03/2008	19/03/2008	360	AdminDrug	PRC	O3D	57.735027
04/03/2008	05/03/2008	24	AdminDrug	CFM	IVP	288.675140
11/03/2008	12/03/2008	24	AdminDrug	CFM	IVP	288.675140
19/03/2008	20/03/2008	24	Response Eval. COPP	Juan		
01/04/2008	16/04/2008	360	Summary COPP			
01/04/2008	02/04/2008	24	AdminDrug	VCR	IV	0.866025
08/04/2008	09/04/2008	24	AdminDrug	VCR	IV	0.866025
01/04/2008	16/04/2008	360	AdminDrug	PRD	O3D	23.094009
01/04/2008	16/04/2008	360	AdminDrug	PRC	O3D	57.735027
01/04/2008	02/04/2008	24	AdminDrug	CFM	IVP	288.675140
08/04/2008	09/04/2008	24	AdminDrug	CFM	IVP	288.675140
16/04/2008	17/04/2008	24	Previous Eval.	Tomas		
29/04/2008	14/05/2008	360	Summary COPP			
29/04/2008	30/04/2008	24	AdminDrug	VCR	IV	0.866025
06/05/2008	07/05/2008	24	AdminDrug	VCR	IV	0.866025
29/04/2008	14/05/2008	360	AdminDrug	PRD	O3D	23.094009
29/04/2008	14/05/2008	360	AdminDrug	PRC	O3D	57.735027
29/04/2008	30/04/2008	24	AdminDrug	CFM	IVP	288.675140
06/05/2008	07/05/2008	24	AdminDrug	CFM	IVP	288.675140
14/05/2008	15/05/2008	24	Remission Eval.	Juan		

FIGURE 9. A therapy plan following the Hodgkin's protocol.

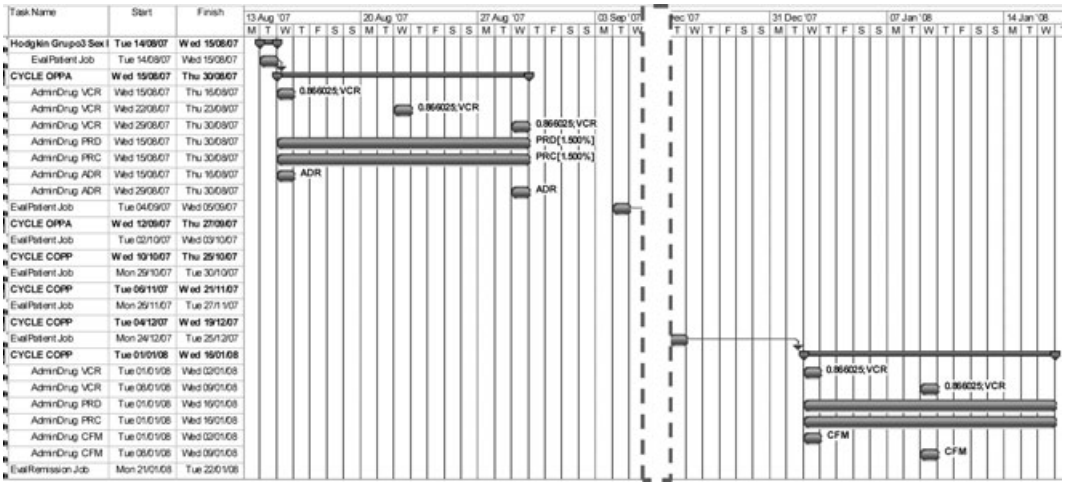

FIGURE 10. A temporally annotated and automatically generated therapy plan represented as a Gantt chart. The plan represents the treatment for a patient of Group1 (male) following the Hodgkin's disease protocol. Start and end dates of every action are shown in the left-hand side. Drugs and their dosage are shown in the bars of the chart.

shown in the figure. It is worth to note that this structure is managed by the planner from the knowledge encoded in the domain, taking advantage of the possibility of encode additional special features in a procedural knowledge representation as the one supported by our planning language. On the other hand, the Gantt chart visualization offers an outline of how these tasks are correctly arranged following the periodic rules of every type of chemotherapy administration.

Contrasted with oncologists, only the generation and visualization in few seconds of a therapy plan (in this case only chemotherapy sessions are displayed) is considered of great help because it saves a lot of time in their current decision-making process. The planner takes less than 1 second to generate a therapy plan, while oncologists take hours to elaborate a therapy plan by hand because they have to consider too many detailed constraints and tasks. Furthermore, at present they are using unstructured, personal text-based templates to write therapy plans, and representing plans in an electronic format as previously shown is also considered as an important advance. In addition, experiments show that the size of plans obtained is considerably smaller than the size of plans in traditional planning applications. For example, in Fdez-Olivares et al. (2006) the same planning technology has been employed to support the generation of fire fighting plans with hundred of actions. However, the size of plans here obtained cannot be considered as a drawback or an obstacle to introduce AI planning technology in this domain. Indeed, the utility of this technology is based on the capability of shifting detailed and repetitive human decisions to an automated planning process. As explained in previous sections, this results in a reduction of the workload of oncologists and, therefore, may lead to improve the quality of health delivery.

This proof of concept has been focused on the validation of the AI planning technology here presented to support both clinical processes and decisions in the field of (pediatrics) oncology therapy planning. At present, a Clinical Decision Support System for oncology therapy planning is being developed in the framework of *OncoTheraper*, a recently started R&D project (funded by the Regional Andalusian Government) participated by our research group together with the pediatrics oncology services (distributed in six different hospitals)

of the Public Health System of Andalusia and two private companies (IActive Intelligent Solutions, a spin-off started up from our research group, and AT4Wireless). The operation of this system is outlined in the following section.

6. OPERATION OF THE SYSTEM

OncoTheraper is intended to be used by oncologists in two decisive steps of the care providing life cycle, namely, treatment planning and treatment monitoring. During the treatment planning step, an oncologist interacts with the therapy planning system (based on the techniques described in previous sections), and consults with the system before each treatment to obtain and validate an initial therapy plan. The main requirement demanded by oncologists about the output of the planner for this step is that therapy plans obtained must follow the guidelines of the treatment protocol as well as cover a time horizon such that all the clinical treatment tasks (chemotherapy, radiotherapy, and evaluation sessions) are included. As shown in the previous section, the therapy plans obtained by the planner adjust to this requirement.

The treatment monitoring step starts once an initial therapy plan has been generated and validated by the oncologist. This step is supported by an execution monitoring system that has been defined in de la Asunción et al. (2007) and applied to a different domain application (Fdez-Olivares et al. 2006) that, nevertheless, shares the same representation of plans. At present, it is being adapted to the new requirements of the application domain object of this work, although the main features of this process are also valid for this application. The monitoring process is designed to interactively support the correct execution of the actions contained in a therapy plan, and it is aimed to guarantee a safe treatment process. Among other functionalities, it might alert clinicians about deviations from the initial plan, forcing oncologists to confirm the beginning of a chemotherapy cycle or avoiding the activation of actions once they have finished.

The monitor receives as input an XML representation of a therapy plan that contains actions that represent clinical activities as well as clinical decisions that an oncologist should follow during the treatment of a patient. These actions are deployed on top of a STN that supports the flexible representation of time intervals that constraint start and end execution times of actions (an example plan with fixed time points is shown in Figure 10). At the beginning of the execution of a therapy plan given as input, its actions, temporal constraints, and facts representing preconditions and effects of actions are consistent with respect to the initial conditions of the planning problem. The execution monitoring process is an event-driven, real-time algorithm that follows the execution of such temporal plans at the highest level of detail because it keeps track of the time at which every effect of every action is achieved. Therefore, it is possible to check that every action executes as predicted. As the plan execution progresses, to support a robust plan execution, the system is capable of detecting inconsistencies that affect either to the temporal dimension of the plan (a delay in the administration of a drug) or to the preconditions of actions (a patient does not progress as expected). The policy applied in such cases is the following:

- In the case of temporal inconsistencies, the execution monitoring process distinguishes between *feasible* delays and *unfeasible* delays. The former ones may also be either local delays (they only affect to an isolated parallel branch of a therapy plan, and thus, a new local reschedule may be found only for that branch) or global delays (they might affect all the remaining actions of the plan, and a whole new reschedule might be needed). In both cases, the rescheduling process is based on the propagation and validation of the

temporal constraints of the plan, following an algorithm that is described in (Castillo et al. 2006).

- In the case of unfeasible delays or violation of the precondition of an action no reschedule is possible, and a new planning/replanning episode, using the above described planning process, is considered.
- Finally, experts may also detect some perturbation in the execution of a plan and, in this case, a new cycle of decision making is started.

7. RELATED WORK

The approach here presented should not be considered only as a new different way to represent therapies. Regarding other approaches devoted to therapy plan management (such as, Asbru, Duftschmid et al. 2002 or Glare, Terenziani et al. 2006), 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. Therefore, the planning 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.

First, the representation and reasoning about temporal constraints of our approach allows to simultaneously validate temporal constraints while generating therapy plans (plan generation, temporal constraint management and reasoning are interleaved). Most approaches (Augusto 2005) are only focused in one side of the problem of therapy planning because they only consider how to manage temporal constraints of actions, and neglect aspects related to how automatically generate sequences of actions with temporal constraints. 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.

Very few (Duftschmid et al. 2002; Votruba et al. 2006) face the problem of plan generation, but it is carried out following a static, nondeliberative process (close to case-based planning), which 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, which affects negatively to the efficiency of the overall process, as well as to important reasoning aspects, such as, 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 Castillo et al., 2006). These features are especially important when plans have to be readapted due to new circumstances arisen during the treatment stage. Furthermore, the process performed by other approaches to temporal constraints verification could be used at execution time to revise possible temporal inconsistencies (such as, 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 (e.g., when a patient's stratification group changes because his/her tumor size does not progress as expected). In such cases our approach might 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.

Considering the planning language here described (a temporal extension of HTN), it should be seen as a knowledge representation mechanism to both, represent human expertise, and use it as a guide to the planning process. Although McDermott's HTN extension of PDDL (McDermott 2003) incorporates expansion methods (that could be used to represent operating procedures), it does not incorporate mechanisms to describe domain heuristics (such as, e.g., *:inline tasks*) that are followed by oncologists to perform tasks and make decisions. Furthermore, our time representation allows to easily encode time constraints on both compound and primitive tasks, as well as to describe synchronization mechanisms between them. The time representation used in McDermott (2003) relies on a semantics of processes and it is based on a sophisticated syntax that is much more complex to encode and manage than the one of our "light" time representation based on STNs.

Finally, it is important to remark that the HTN domain of this approach does not only captures procedural knowledge (as already explained the planning language has the capability of representing common workflow patterns) but also declarative knowledge: the planning representation supports the declarative description of alternative courses of actions and different subgoaling strategies (both supported by decomposition schemes) as well as qualitative and quantitative scheduling constraints (causal and order relationships, deadline goals, or resource availability constraints) in several knowledge structures (deductive rules, PDDL actions, HTN tasks, and temporal constraints at different levels of abstraction). These are different types of domain knowledge that require the integration of different techniques (HTN planning, forward search, deliberative, and temporal reasoning) in the planning process.

In this sense, the first stages of the treatment planning decision-making process (devoted to generate an initial, temptative therapy plan to support the initial decision-making process of oncologists) require a temporal HTN planning process based on deliberative techniques, such as, the one here shown (similar to other real-world planning systems, such as, O-PLAN (Tate, Drabble, and Kirby, 1994), or SIPE (Wilkins and desJardins, 2001). These techniques present a better operation and expressiveness for this purpose than specific procedural reasoning systems, such as, PRS (Georgeff and Lansky 1987) or PRS-CL (Wilkins and Myers 1995). These systems are intended to represent only procedural knowledge (concretely collections of structured actions for use in specific situations) and they are oriented to support a reactive, event-driven behavior. As these systems have been traditionally used to support design of real-time, continuously active intelligent systems, they are much more appropriate to support the execution monitoring of therapy plans instead of the knowledge-based generation of plans.

8. CONCLUSIONS

In this work, we have presented an AI P&S system based on temporal HTN that provides support for both representing clinical processes and making clinical decisions. The HTN planning language (a temporal and hierarchical extension of PDDL) and the hierarchical planning and scheduling process are 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. In addition, as shown in the experiments, the guidelines provided in such protocols to administrate drugs are represented in a simple, intensional mode, which together with the deliberative, temporal-based knowledge-driven process, allows to generate extensional schedules for each chemotherapy cycle.

From the health assistance point of view, this approach presents some benefits: on the one hand, oncologists recognize that their workload might be reduced in benefit of the patient (they spend hours in planning an accurate therapy while our system obtains the same therapy

plans in few seconds), which allow to improve health delivery quality. On the other hand, authors argue that patient safety might be augmented because the recommended actions to administrate drugs are based on an automated planning process. The opinion of oncologists (when they were interviewed in the plan validation stage) is that the automated validation of temporal constraints may help to reduce administration errors related with time management, as well as the management of numerical resources might reduce dosage errors. However, at present the system is not deployed and we cannot contribute with real data about the impact of this tool on the error rate in clinical practice. This is a subject of further study. In addition, the electronic representation and operationalization of clinical oncology protocols (supported by an appropriate distributed architecture) helps to access the knowledge of the protocols from any place at any time, thus the representation of human resources in therapy plans may also help to assign resources to tasks involved in the treatment, improving resource coordination and reducing communication errors (everybody knows what task to do and when to do it).

Results shown in this work should be considered as the first step in the process of the full development and deployment of a Clinical Decision Support System for therapy planning based on oncology treatment protocols. Many aspects in this sense are at present subject of further development in the framework of *OncoTheraper*. One of the main challenges is related to the integration with legacy clinical information systems already existing in the working environment of oncologists, particularly those related with both how to translate the information stored in EHRs into the planning representation language, and how to store the plans obtained by the planner in EHRs following a standard representation for clinical information. Most hospitals incorporate in their clinical information systems a *drug stock control system* that is already being used by oncologists to send drug-administration orders. Indeed, the opinion of oncologists is that the output of the therapy planning system here presented might be of great help if it were integrated with the input to such system.

Finally, we cannot neglect the use of knowledge engineering techniques to support the process of representing oncology protocols in our planning language. The proliferation of standard languages and frameworks for modeling and editing CIGs (Peleg et al. 2003) is well known. As explained in the introduction and shown through this article, our planning language embodies most of the features of such languages. 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.

ACKNOWLEDGMENT

This work has been partially supported by the Andalusian Regional Ministry of Innovation under project P08-TIC-3572.

REFERENCES

AUGUSTO, J. 2005. Temporal reasoning for decision support in medicine. Artificial Intelligence in Medicine, **33**:1–24.

BRADBROOK, K., G. WINSTANLEY, D. GLASSPOOL, J. FOX, and R. GRIFFITHS. 2005. Ai planning technology as a component of computerised clinical practice guidelines. Lecture Notes in Artificial Intelligence, **3581**:171–180.

BRESINA, J. L., A. K. JONSSON, P. MORRIS, and K. RAJAN, 2005. Activity planning for the mars exploration rovers. *In* Proceedings of the ICAPS05, Monterey, CA, pp. 40–49.

CASTILLO, L., J. FDEZ-OLIVARES, O. GARCÍA-PÉREZ, and F. PALAO. 2006. Efficiently handling temporal knowledge in an HTN planner. *In* Proceeding of ICAPS06, Lake District, Cumbria, UK, pp. 63–72.

CASTILLO, L., J. FDEZ-OLIVARES, O. GARCÍA-PÉREZ, T. GARZÓN, and F. PALAO. 2007. Reducing the impact of AI planning on end users. *In* ICAPS 2007, Workshop on Moving Planning and Scheduling Systems into the Real World, Providence, RI, pp. 40–49.

DE LA ASUNCIÓN, M., L. CASTILLO, J. FDEZ-OLIVARES, O. GARCÍA-PÉREZ, A. GONZÁLEZ, and F. PALAO. 2007. Handling fuzzy temporal constraints in a planning framework. Annals of Operations Research on Personnel Scheduling and Planning, **155**(1):391–415.

DUFTSCHMID, G., S. MIKSCH, and W. GALL 2002. Verification of temporal scheduling constraints in clinical practice guidelines. Artificial Intelligence in Medicine, **38**(2):171–195.

FDEZ-OLIVARES, J., L. CASTILLO, O. GARCÍA-PÉREZ, and F. PALAO. 2006. Bringing users and planning technology together. Experiences in SIADEX. *In* Proceedings ICAPS06, Monterey, CA, pp. 11–20.

GEORGEFF, M., and A. LANSKY. 1987. Reactive reasoning and planning. *In* AAAI 87, Seattle, WA, pp. 677–682.

GHALLAB, M., D. NAU, and P. TRAVERSO. 2006. Automated Planning. Theory and Practice. Morgan Kaufmann, Amsterdam, the Netherlands.

GROUP, S. W. 2003. National protocol for diagnosis and treatment of Hodgkin's disease (in Spanish).eh-seop.003. Technical report, SEOP.

LEONG, T., K. KAISER, and S. MIKSCH. 2007. Free and open source enabling technologies for patient-centric, guideline-based clinical decision support: A survey. Methods of Information in Medicine, **46**:74–86.

MCDERMOTT, D. 2003. The formal semantics of processes in PDDL. *In* ICAPS 2003. Workshop on PDDL, Trento, Italy.

MULYAR, N., W. M. VAN DER AALST, and M. PELEG. 2007. A pattern-based analysis of clinical computer-interpretable guideline modeling languages. Journal of American Medical Information Association, **14**(6):781–787.

PELEG, M., S. TU, J. BURY, P. CICCARESE, J. FOX, and R. G. RA. 2003. Comparing computer-interpretable guideline models: A case-study approach. Journal of American Medical Information Association, **10**:10–58.

SACERDOTI, E.D. 1975. The nonlinear nature of plans. *In* Proceedings of IJCAI 1975, Tbilisi, Georgia, USSR, pp. 206–214.

SPYROPOULOS, C. 2000. AI planning and scheduling in the medical hospital environment. Artificial Intelligence in Medicine, **20**:101–111.

TATE, A. 1977. Generating project networks. *In* Proceedings of IJCAI 1977, Cambridge, MA, pp. 888–893.

TATE, A., B. DRABBLE, and R. KIRBY. 1994. O-PLAN2: An open architecture for command, planning and control. *In* Intelligent Scheduling. *Edited by* M. Zweben and M. Fox. Morgan Kaufmann, Amsterdam, the Netherlands.

TERENZIANI, P., L. ANSELMA, S. MONTANI, and A. BOTTRIGHI. 2006. Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. Artificial Intelligence in Medicine, **38**(2):171–195.

VOTRUBA, P., A. SEYFANG, M. PAESOLD, and S. MIKSCH. 2006. Environment-driven skeletal plan execution for the medical domain. *In* Proceedings ECAI2006, Trento, Italy, pp. 847–848.

WILKINS, D. E., and M. DESJARDINS. 2001. A call for knowledge-based planning. AI Magazine, **22**(1):99–115.

WILKINS, D. E., and K. L. MYERS. 1995. A common knowledge representation for plan generation and reactive execution. Journal of Logic and Computation, **4**(6):731–761.