Supporting clinical processes and decisions by hierarchical planning and scheduling

Juan Fdez-Olivares and **Luis Castillo**
Department of Computer Science and A.I
University of Granada, Spain
email: {faro,L.Castillo}@decsai.ugr.es

Juan A. Cózar
Pediatrics Oncology Service
Hospitalary Complex of Jaén, Spain
email: jcozarolmo@hotmail.com

Oscar García Pérez
IActive Intelligent Solutions
`http:://www.iactive.es`
email: o.garcia@iactive.es

Abstract

This paper is focused on how a general-purpose hierarchical planning representation, based on the 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.

MOTIVATION

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, by means of a reasoning process driven by the procedural knowledge encoded in the HTN domain, in order to determine how to perform a high-level task introduced as problem. This planning paradigm, from a practical point of view, cannot 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 (Castillo et al. 2007; Bresina et al. 2005), particularly 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 (there is an 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 (Augusto 2005; Votruba et al. 2006; Spyropoulos 2000), which are aimed to recommend predefined general courses of action to be applied to a patient, on the process of treating a disease.

These systems incorporate, on the one hand, a computerized representation of clinical protocols, also called *Computer Interpretable Clinical Guidelines* (CIGs)(Peleg et al. 2003): 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 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, Kaiser, and Miksch 2007), all of them 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 adequate technique that might support both the representation of clinical processes and clinical decision making in therapy planning, by taking advantage of its deliberative and knowledge driven reasoning process to automatically generate treatment plans, starting from an accurate representation of clinical protocols. However, up 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, Miksch, and Gall 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.

Thus 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 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 into the reasoning process in order 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).

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 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 (Mulyar, van der Aalst, and Peleg 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. Monitoring 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 (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 state. 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, thought it is possible to access patient's medical information in Electronic Health Records, 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 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.

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 (Fdez-Olivares et al. 2006). It uses as its planning domain and problem description language an HTN extension of PDDL, a language used by most of well known *primitive action-based* 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 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 (Castillo et al. 2006) 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)

```
(: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*.

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 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.

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.

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 (Leong, Kaiser, and Miksch 2007; Peleg et al. 2003)). 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 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 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.

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 (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 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 (Terenziani et al. 2006)).

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 (Augusto 2005) are only focused in one side of the problem of therapy planning, since they only consider how to manage temporal constraints of actions, and neglect aspects related to how automatically generate sequences of actions with temporal constraints. Very few (Duftschmid, Miksch, and Gall 2002; Votruba et al. 2006) 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

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

```
(: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.

the plan as a guide to propagate constraints (as is the case of our planner (Castillo et al. 2006)). These features are specially important when plans have to be readapted due to new circumstances arisen during the treatment stage.

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.

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 (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 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

```
(: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

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.

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 in order 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, since 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`.

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.

PROOF OF CONCEPT

Considering the previous description, a proof of concept of this technology has been carried out in collaboration with

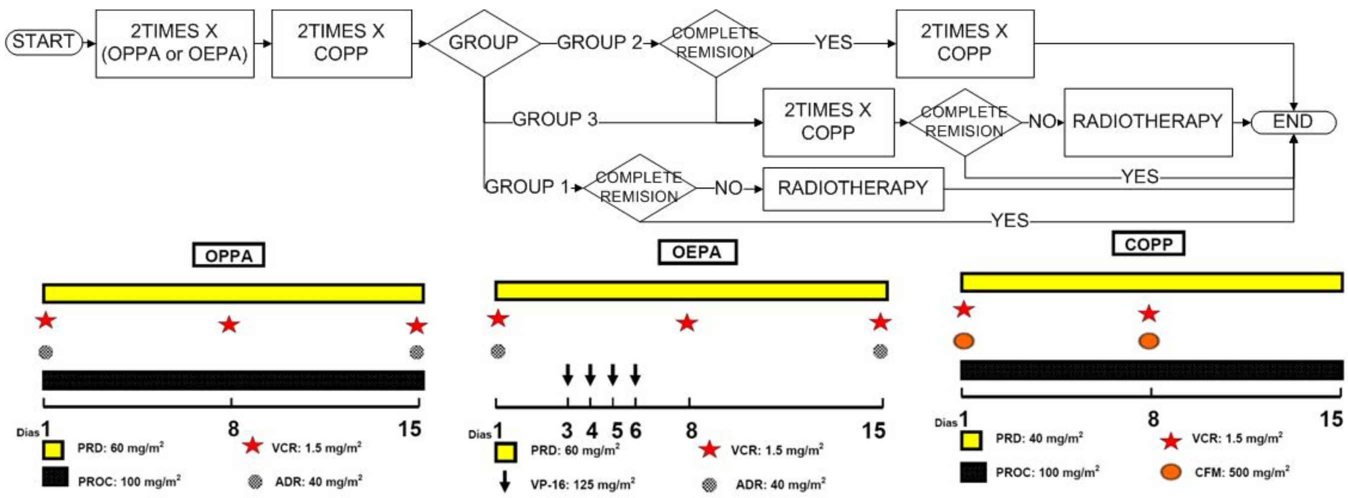

Figure 4: A general schema of Hodgkin's Disease Clinical Protocol. The representation followed to show the periodical temporal patterns for chemotherapy cycles (OPPA, OEPA and COPP) is literally copied from the protocol specification.

```
(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

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 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 flowchart diagram of Figure 4. 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 tumour 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 tumour is not detected.

Temporal patterns to administrate every type of chemotherapy cycle are shown below the flow-chart of Figure 4. For example, a cycle of type OPPA takes 15 days, the rules to administrate a cycle of type OPPA state that the drugs PRD and PROC 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.

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 in a domain file. The domain includes six compound tasks, 13 methods, 6 primitive tasks and the file contains more than 400 lines of code [2].

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 6). Several experiments were realized on different patient profiles, and all the plans were obtained in less than one second. 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.

The plan shown in Figure 6 has been obtained after a postprocessing of the output of the planner, in order to friendly

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

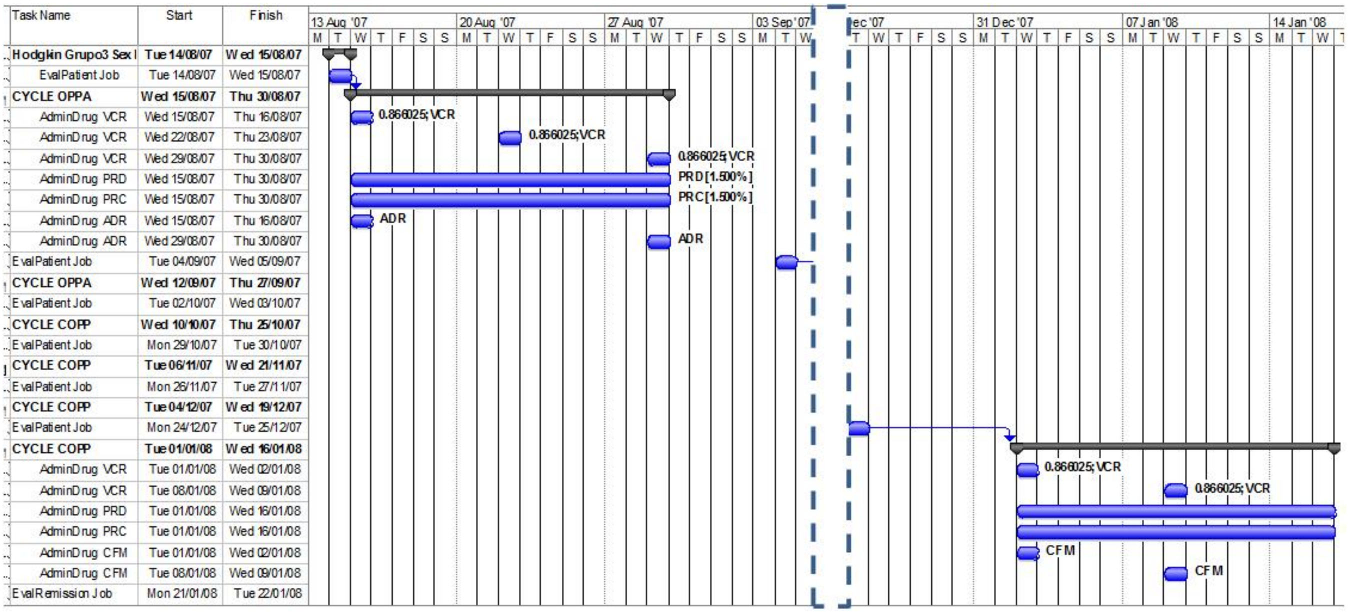

Figure 6: A temporally annotated and automatically generated therapy plan represented as a gantt chart. The plan represents the treatment for a patient of Group1 (male) following the Hodgkin's Disease Protocol. Start and end dates of every action are shown in the left-hand side. Drugs and their dosage are shown in the bars of the chart.

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 taks* as AdminDrug), that may be collapsed or deployed as shown in the figure. It is worth to note that this structure is managed from the planner and 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 Gant 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 visualized) is considered of great help, since it saves a lot of time in their current decision making process, because oncologist have to take into account too many detailed constraints and tasks that, by the other way, we have shown to be accurately represented in a temporally extended HTN representation.

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 (shown in the left hand side of the Figure 6. Therefore, at the beginning of the execution 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 (Fdez-Olivares et al. 2006) 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.

CONCLUSIONS

In this work we have presented an AI P&S system based on temporal Hierarchical Task Networks (HTN) that provides support for both representing clinical processes and making clinical decisions. The HTN planning language (a 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. Our planning language should be seen as a knowledge representation mechanism to represent human expertise and to 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 (like, for example, *:inline tasks*) that are followed by oncologists in order 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 Simple Temporal Networks.

This approach should not be considered only as a new way to represent therapies. Regarding other approaches devoted to therapy plan management (like Asbru (Duftschmid, Miksch, and Gall 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 lifecycle 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 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.

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 planing based on oncology treatment protocols. We cannot neglect the use of knowledge engineering techniques in order to support the process of representing oncology protocols in our planning language. It is well known the proliferation of standard languages and frameworks for modeling and editing CIGs (Peleg et al. 2003). As explained in the introduction and shown thorough this paper 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.

References

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

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

Bresina, J. L.; Jonsson, A. K.; Morris, P.; and Rajan, K. 2005. Activity planning for the mars exploration rovers. In *Proceedings of the ICAPS05*, 40–49.

Castillo, L.; Fdez-Olivares, J.; García-Pérez, O.; and Palao, F. 2006. Efficiently handling temporal knowledge in an HTN planner. In *Proc. ICAPS*.

Castillo, L.; Fdez-Olivares, J.; Garca-Prez, O.; Garzón, T.; and Palao, F. 2007. Reducing the impact of ai planning on end users. In *ICAPS 2007, Workshop on Moving Planning and Scheduling Systems into the Real World*.

Duftschmid, G.; Miksch, S.; and Gall, W. 2002. Verification of temporal scheduling constraints in clinical practice guidelines. *Artificial Intelligence in Medicine*.

Fdez-Olivares, J.; Castillo, L.; García-Pérez, O.; and Palao, F. 2006. Bringing users and planning technology together. Experiences in SIADEX. In *Proc. ICAPS*.

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

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

Leong, T.; Kaiser, K.; and Miksch, S. 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*.

Mulyar, N.; van der Aalst, W. M.; and Peleg, M. 2007. A pattern-based analysis of clinical computer-interpretable guideline modeling languages. *JAMIA*.

Peleg, M.; Tu, S.; Bury, J.; Ciccarese, P.; Fox, J.; and RA, R. G. 2003. Comparing computer-interpretable guideline models: A case-study approach. *J Am Med Inform Assoc* 10:10–58.

Sacerdoti, E. D. 1975. The nonlinear nature of plans. In *IJCAI 1975*, 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 *IJCAI 1977*, 888–893.

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

Votruba, P.; Seyfang, A.; Paesold, M.; and Miksch, S. 2006. Environment-driven skeletal plan execution for the medical domain. In *ECAI-2006*, 847–848.