Careflow Planning: From Time-Annotated Clinical Guidelines to Temporal Hierarchical Task Networks*

Arturo González-Ferrer[1], Annette ten Teije[2],
Juan Fdez-Olivares[1], and Krystyna Milian[2]

[1] Department of Computer Science and AI, Universidad de Granada, Spain
[2] Department of Computer Science, Free University Amsterdam, The Netherlands

Abstract. Decision-making, care planning and adaptation of treatment are important aspects of the work of clinicians, that can clearly benefit from IT support. Clinical Practice Guidelines (CPG) languages provide formalisms for specifying knowledge related to such tasks, such as decision criteria and time-oriented aspects of the patient treatment. In these CPG languages, little research has been directed to efficiently deal with the integration of temporal and resource constraints, for the purpose of generating patient tailored treatment plans, i.e. care pathways. This paper presents an AI-based knowledge engineering methodology to develop, model, and operationalize care pathways, providing computer-aided support for the planning, visualization and execution of the patient treatment. This is achieved by translating time-annotated Asbru CPG's into temporal HTN planning domains. The proposed methodology is illustrated through a case study based on Hodgkin's disease.

1 Introduction

Care pathways are increasingly seen as a means to put Clinical Guidelines in practice by interdisciplinary teams; they help to reduce patient uncertainty and delays, to improve resource utilization and enhance efficiency savings, and to develop a family-centred care [12]. Specifically, the aim of care pathways is to model a timed process of patient-focused care, by specifying key events, clinical exams and assessments to produce the best prescribed outcomes, within the limits of the resources available, for an appropriate episode of care [1]. While these pathways were not traditionally embedded into IT-supported environments, the new trend is to code organizational arrangements into systems such as scheduling and workflow engines [16]. So, automated guideline-based generation of care pathways is clearly of interest in the clinical domain, but its resolution is not trivial, in particular when considering the existence of temporal constraints.

Indeed, the management of uncertainty and temporal issues is highly relevant in the CPG's domain [2], and some research goals were identified recently,

* Work partially supported by projects P08-TIC-3572 and TIN2008-06701-C03-02. We gratefully acknowledge the assistance and proof-reading by Frank van Harmelen.

M. Peleg, N. Lavrač, and C. Combi (Eds.): AIME 2011, LNAI 6747, pp. 265–275, 2011.

specifically for the treatment of temporal aspects [19]. Concretely, one of these goals is the identification of candidate actions for the care process, highly related to the automated generation of care pathways. Nonetheless, the research in AI has shown that the complexity of temporal inference is strictly related to the expressivity of temporal languages [19]. Specifically, both CPG languages and Planning & Scheduling (P&S) languages have been shown to be very expressive in terms of their temporal constraints representation [19,4], but they provide different capabilities. While CPG languages are very user-friendly for the guideline acquisition phase, one of the main shortcomings of the associated execution engines developed so far [11], is that they do not provide support for the generation of end-to-end tailored treatment plans. On the other hand, traditional P&S languages and techniques have shown their potential for representing and interpreting temporal information, allowing the automated generation of time-annotated plans, the adaptation to custom constraints and the allocation of resources to tasks [6].

Hence, on the basis of the existing similarities between some CPG's representational structure and the Hierarchical Task Network planning paradigm [5,4], we propose in this paper the translation of the knowledge present in a CPG into a corresponding temporal HTN planning domain. Thus, starting from an automated knowledge acquisition process where the CPG temporal information is analyzed and translated into the corresponding HTN domain, and followed by a knowledge-driven process based on P&S techniques, we can obtain a situated plan for the steps to be carried out for the patient's treatment. This way, we can automatically generate a care pathway that can later be deployed into any executable form (e.g. a workflow engine for plan visualization and execution), providing a cornerstone for the development of guideline-based careflow management systems [18].

The paper is structured as follows. Section 2 introduces the languages used and the methodology followed, section 3 describes the representation of temporal patterns in both languages, section 4 shows the rules used for the translation, section 5 shows some results and section 6 offers some conclusions.

2 Materials and Methods

In this section, a brief overview of the CIG and P&S languages used is given, as well as the life-cycle of the methodology presented.

CIG languages: Asbru. Different Computer Interpretable Guideline languages (CIG's) were developed in recent years, with the aim to manage multiple modeling aspects of guidelines [17,11]. Since the translation of temporal patterns is one of our main research aims, the methodology here presented is applied to Asbru[14], given its known capacity to model time-oriented aspects. Furthermore, Asbru follows a hierarchical decomposition of guidelines into networks of component tasks that unfold over time, known as Task Network Model (TNM), that is also followed by the target HTN planning language, described later. Fortunately, many of the existing CIG languages are TNM-based as well, so the methodology could easily be adapted to any other of these languages, since they share the organization of plan components, and they can express multiple arrangements of these components and interrelationships between them [17].

Asbru is an XML-based task-specific, time-oriented and intention-based plan representation language to embody CPG's and protocols as skeletal plans [13]. Each one of these skeletal plans consists of a *plan-body* that can be composed of either *subplans* (a set of steps performed in parallel or sequentially), *cyclical-plan* (repeated several times), *plan-activations* (a call to another plan) or a *user-performed* step (a specific action performed by the user). In addition, time-annotated conditions can be attached for the selection of plans, or the transition of a plan between multiple states. The tightly coupled control loop between the generation and execution of plans (i.e. continual planning [13]), makes Asbru very compelling for the management of CPGs, specifically in high-frequency domains. Even so, our aim is to support low-frequency domains, where the introduction of workflow capabilities could be highly interesting, in order to carry out the human-centered execution of long term care pathways. Also, note that what is relevant for this paper are the powerful knowledge representation capabilities of Asbru, not its execution engine.

Hierarchical Task Network planning: HPDL. The HPDL planning language (described in [4]) is an HTN [5] extension of the well-known planning language PDDL [7]. HTN planning specifications are separated into a planning *domain* (predicates, actions and tasks), designed as a hierarchy of tasks representing compound and atomic activities (see Figure 2), and a planning *problem* where objects, initial states and a set of goals are outlined. In such hierarchical domains, it is possible to describe how every compound task may be decomposed into different subtasks and the order that subtasks must follow, by using different *methods*. These methods include a precondition that must be satisfied by the world state in order for the method to be applicable by the planner.

HTN planning is known to be very useful in real-world applications. A recent study[2] described how different planning paradigms, including HTN, can provide support for the execution of CPG's. Actually, it has shown to be an enabling technology to support clinical decisions and processes in medical treatments[6].

The methodology presented in this paper can be observed in Figure 1. Specifically, we modeled in Asbru the Hodgkin disease guideline, where the goal of an

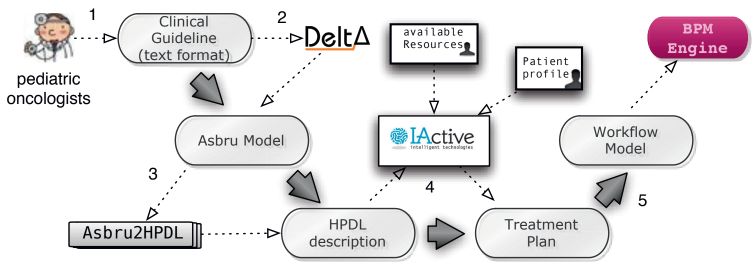


Fig. 1. Methodology life-cycle. Numbers 1-5 represent the multiple steps.

oncologist when planning a treatment, is mainly to schedule chemotherapy, radiotherapy and monitoring of a patient. Even so, Asbru guidelines are conceived to be directly executed, but they are not conceived to support the automated generation of patient-focused care pathways. For this reason, we translate the Asbru model into a corresponding HTN planning domain, enabling, by means of a knowledge-driven process, the automated generation of these care pathways. The pathways can then be deployed in the form of a *careflow*, thus offering an ubiquitous, user-friendly access (e.g. mobile or PDA-based) to a workflow engine, providing a communication channel for the care team. This facilitates the execution of every step in the pathway, checking, through customized triggers correctly embedded into the workflow engine, that the steps are carried out following the temporal patterns initially specified in the original guideline [8].

Note that Asbru models can be translated into HPDL domains because of the structural similarities between them: a) Asbru skeletal plans are equivalent to HPDL compound tasks or primitive actions, b) both share a hierarchical structure c) both are able to represent several different task ordering schemas, and d) both are powerful and expressive for the representation and interpretation of temporal constraints. The next section is devoted to the representation of such constraints, while the steps depicted in Figure 1 are analysed in section 5.

3 Analysis of Temporal Knowledge

The aim of this section is to show how to represent Asbru's temporal contraints with HPDL, in order to introduce the translation algorithm presented later in section 4. The representation with HPDL of Asbru's multiple task ordering schemas (parallel, sequential, unordered or any-order), commonly known as basic *workflow patterns* [15], has been already studied in [4], and also for the translation of business process models in [9]. Therefore, our analysis is focused specifically on time annotations, synchronization or delays among tasks and repetitive or cyclical temporal patterns.

Time annotations. Asbru time annotations are used to constrain the temporal occurrence of plan elements (including the plans themselves). A time annotation can include three time ranges, constraining start and end times and the duration of the interval. These constraints are defined as time shifts relative to a reference point, enabling to easily define them relative to an event not known at plan creation time (e.g. the start of a plan). The plan starts in a specific *starting interval* [ESS, LSS] and finishes in a *finishing interval* [EFS, LFS]. Furthermore, its duration has to be within the *duration interval* [minDur, maxDur].

Asbru time annotations can be equivalently represented with the HPDL temporal annotations that are described in [4]. In the case of HPDL, every primitive action (or task) a_i has two time points $start(a_i)$ and $end(a_i)$. Therefore, given a time-annotated Asbru skeletal plan T (remember that an Asbru model is compound of several skeletal plans, and that these skeletal plans are equivalent to HPDL tasks or actions), we can express it with an equivalent HPDL task or action T' by using special time-point variables `?start`, `?end` and `?duration`:

`[ESS,LSS]` for T can be expressed in HPDL with `(and (>= ?start ESS) (<= ?start LSS))`, `[EFS,LFS]` can be expressed as `(and (>= ?end EFS) (<= ?end LFS))`and, finally, `[minDur,maxDur]` as `(and (>= ?duration minDur)(<= ?duration maxDur))`.

A more complex issue is the management of the reference time points that can appear in an Asbru model: a fixed point in time, a reference to a plan activation, or a set of cyclical time points. We focus first on the last two.

Time-annotated reference to plan activation. This mechanism allows us to synchronize the timeline of skeletal plans. For example, to synchronize a plan B with the start of a plan A, B can refer to "A entering activated state", while synchronizing with the end of a plan A could be done with a reference to "A leaving complete state". Furthermore, a delay can be added with respect to the referenced plan. These are known as plan-pointers in Asbru, and they can be similarly represented by means of so-called *temporal landmarks*[4] in HPDL. These landmarks are asserted in the current state, and later on, they may be recovered and posted as deadlines to other tasks in order to synchronize two or more activities. This is done by means of *deductive inference tasks* of the form `(:inline <p><c>)`, fired when the 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 expression `<c>`. Figure 2(a) shows the definition of a compound task representing the parallel execution of four drug administration cycles. This task is executed sequentially two times, where the second is delayed 28 days since the start of the first one.

```
(:task CommonInit
 :parameters(?p - patient)
 (:method boys
  :precondition()
  :tasks(
    (CYCLE-OPPA ?p)
    ((and (>= ?start (cstart OPPA))
    (= ?duration (* 28 24)) ) (Delay OPPA))
    ((>= ?start (last-completed OPPA))
    (CYCLE-OPPA ?p))))
)
(:task CYCLE-OPPA
 :parameters(?p - patient)
 (:method only
  :precondition()
  :tasks(
    (:inline () (assign (cstart OPPA) ?start))
    [(GenericCycle ?p AdminVCR OPPA 1.5 0)
     (GenericCycle ?p AdminPRD OPPA 60 0)
     (GenericCycle ?p AdminPRC OPPA 100 0)
     (GenericCycle ?p AdminADR OPPA 40 0)
    ])))
```

(a)

```
(:task GenericCycle
 :parameters (?p-patient ?c-Ciclo ?cq-CicloQ ?dose ?N)
 (:method prepare
  :precondition (and (< ?N (NRep ?c ?cq))(not (started ?c ?cq))
                (fromStart ?c ?cq ?d)(bind ?N 0))
  :tasks  ((:inline () (assign (last_iter_end ?c) ?start))
           (:inline () (started ?c ?cq))
           (GenericCycle ?p ?c ?cq ?dose ?N))
  )
  (:method loop
   :precondition (and (started ?c ?cq) (< ?N (NRep ?c ?cq))
                 (RepDur ?c ?cq ?d)
                 (fromStart ?c ?cq ?d))
   :tasks((:inline (or
         (and (inBetween ?c ?cq 0)(bind ?K (NRep ?c ?cq)))
         (and (not (inBetween ?c ?cq 0))(bind ?K (+ ?N 1))))())
         (GenericCycle ?p ?c ?cq ?dose ?K))
   )
   (:method stop
    :precondition (>= ?N (NRep ?c ?cq))
    :tasks ((:inline () (not (started ?c ?cq)))))
)
```

(b)

Fig. 2. (a) use of temporal landmarks and parallel execution of several cycles for drug administration, (b) outline of a recursive HPDL generic cyclical task

Cyclical time annotations. Besides temporal annotations, Asbru also has specific temporal semantics for cyclical plans, called *cyclical time annotations*. The difference with the former is a more complex specification for the reference time point consisting of a *time point*, an *offset* and a *frequency*. Also, a *times-completed* condition can be specified in order to determine the number of repetitions for the plan. We explain next how to represent cyclical tasks with

HPDL (see Figure 2(b)). To that end, we have used a high-level formalism, developed by Anselma et. al [3]. Concretely, this formalism shares some similarities with Asbru cyclical time annotations and it can be represented in HPDL as well. The following primitives are considered at the moment:

1. **fromStart**(min, max). Represents a delay between the start of the timespan where actions are to be included and the beginning of the first repetition. It is similar to Asbru's `<offset>`. In HPDL we add a predicate `(fromStart ?t ?c ?o)`, encoding that task *t* has offset *o* with respect to the start of cycle *c*.
2. **inBetween**(min, max). Represents a delay between the end of each repetition and the start of the next one. It is similar to Asbru's `<frequency>` element. In HPDL we can add a predicate `(inBetween ?t ?c ?f)`, encoding that task *t* has a repetition frequency of *f*, within cycle *c*.
3. **NRep**. Represents the number of repetitions to be carried out. It is similar to Asbru's `<times-completed>` element. In HPDL we can add a function `(NRep ?t ?c)`, encoding that task *t* repeats *n* times, within cycle *c*.
4. **RepDur**. Represents the time that each repetition takes. It is similar to Asbru's `<duration>` element. In HPDL we can add a predicate `(RepDur ?t ?c ?lt)`, encoding that task *t* within cycle *c* has a duration *lt*.

These HPDL predicates and functions are used in order to represent cyclical plans, by using them in the definition of a cyclical task (see Figure 2(b)), that is instantiated with initial state values, as explained in the next section. Thus, we have shown that main Asbru time annotations can be modelled in HPDL.

4 Mapping Asbru to HPDL

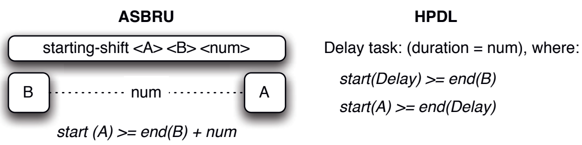
This section shows that the Asbru time modeling constructs have a counterpart HTN representation, supported by the Anselma et. al formalism [3]. We describe a knowledge engineering method to extract a corresponding HTN domain from a typical CPG, using a specific subset of the Asbru language:

Objects and Types. The data model of both languages is very similar. While HPDL types are arranged as a hierarchy where "object" is the upper node, and described in a section called `''(:types"`, Asbru types are defined by means of the element `scale-def`. Thus, we translate every qualitative `scale-def` as an HPDL type, so that each of its `qualitative-entry`'s is declared as a constant in the domain and a corresponding object of that type in the problem.

Skeletal plans. For every Asbru "plan", starting in the root node and traversing the Asbru domain definition hierarchically, use the plan name for the definition of a corresponding HPDL compound task (if it contains plan activations) or durative action (if it is a leaf node, usually "user-performed") in the planning domain. If the plan has any "argument" in its definition, translate it as a HPDL task parameter (e.g. "?d - drug"). In addition, apply the next rules:

1. If the "plan-body" is composed of "subplans" of type *"sequentially"* and it is a sequence of "plan-activations", translate them as a sequence of tasks calls, by enclosing them with `( )`.
2. If it is composed of "subplans" of type *"parallel"*, translate them as a parallel execution of tasks calls, by enclosing them with `[ ]`.
3. If it is composed of an "if-then-else" block, where an argument is evaluated and different plan-activations are carried out for every argument value, this will be translated as several HPDL *task methods*, where every method precondition is the one defined on every "if" statement, and the method body is a call to the corresponding HPDL task (the "plan-schema" activated).

Time annotations in plan activations. When a plan activation is time-annotated, using a reference to a plan-pointer as explained previously, we can express the shifts regarding the pointed plan by using a Delay task in HPDL. For example, every earliest starting-shift can be translated as a Delay from the reference-point specified (the start or end of another skeletal plan) (Figure 3). Thus, if the "instance-type" attribute is 'last' and "state" is 'completed', we will add to the HPDL task referenced by the plan-pointer a temporal landmark (e.g.`(assign (last-completed ?planname) ?end)`). Similarly, if "state" is 'activated', we can define a temporal landmark (e.g. `(assign (last-start ?planname) ?start)`). These landmarks are used in the definition of the Delay task used (Figure 3).




```
;; definition of a Delay task in HPDL
(:task Delay
    :parameters (?pname)
    (:method only
     :precondition ()
     :tasks ( (:inline () (assign (last-completed ?pname) ?end)))
```

Fig. 3. Expressing delays in Asbru and HPDL

Cyclical Plans. Finally, when a plan is defined with a "cyclical-plan-body", we considered that it is declared with at least some arguments: *freq, nrep, fromstart, mindur, maxdur, partof*. Furthermore, we include *drug* and *dose*, since they are described in the Hodgkin CPG. Thus, on the one hand, the cyclical plan can be translated as a HPDL task similar to the one declared in Figure 2(b). On the other hand, we extract several predicate instances (following the temporal formalism explained in the previous section) from these arguments values found in every call to the cyclical plan, so that those instances are serialized into the HTN problem definition, and used correspondingly in the definition of the HPDL cyclical task (see Figure 2(b)). For example, we will extract the predicate `(NRep Prednisona OPPA 15)`, from a call where *nrep*=15, *drug*=prednisona and *partof*=OPPA, in order to express that the drug is administered 15 times within the chemotherapy cycle named OPPA.

5 Results

In order to show the contribution of our method, we carried out the multiple steps of the life-cycle shown in Figure 1 for the Hodgkin protocol. A schema of the treatment workflow process indicated in such protocol, as well as the temporal patterns to administrate every chemotherapy cycle are outlined in [6].

So, looking at Figure 1, the clinical guideline is firstly described in natural language (70 pages of text) starting from the oncologist's experience (1). Then, it is modeled with DELT/A (the Document Exploration and Linking Tool), obtaining an XML-based Asbru model of the CPG (2). Afterwards, a translation to HPDL is necessary (3). We developed a Java tool to that end named Asbru2HPDL (available at `http://gitorious.org/asbru2hpdl`), that carries out two basic steps: (a) the acquisition of the knowledge present in the Asbru CPG by parsing it into a memory structure (a set of java classes), (b) the serialization of this memory structure into an HPDL domain and problem (section 4 is devoted to describe this step). Following with the life-cycle, the IACTIVE planner[1] takes the HPDL domain and problem as input and generates a tailored care pathway for the patient (4). This plan can either be visualized as a Gantt diagram or translated into a workflow instance that can be executed in a BPM engine (5), ideal for environments where doctors have to carry out the treatment in a collaborative way. This last step can be done in order to provide an execution model of the plan, given that the planner does not include yet an embedded execution and monitoring engine. This is being developed at the moment of writing this paper.

Thus, we obtain a planning domain which includes a high-level task goal, subsequently decomposed by the planner following the strategy and temporal constraints declared in the HTN domain, previously translated from the original CPG. Even so, the workflow specified in an oncology treatment protocol does not include details related to which human and material resources are involved in the therapy planning process, so it is necessary to represent and manage them in order to truly support clinical processes and decisions within a specific institution. Fortunately, capacity and availability dates of discrete resources may be represented in HPDL, using a generalization of timed initial literals [4]. Thus, we add to this domain: a) an initial state representing information that describes the patient profile (e.g. (sex Alice M), (group Alice Group3)), and b) resource constraints, mainly related to the oncologists' availability schedule (e.g. (between "08/11/2010 00:00:00" and "09/11/2010 00:00:00" (available John))). Having this information, the planner can find a solution for the problem of obtaining a plan tailored to a patient profile, while respecting the available resources [6].

[1] It has already been used to automatically generate oncology treatment plans [6].

Although we tried to obtain a treatment plan using directly the only freely available Asbru interpreter, we found several problems. The interpreter is prepared to run in high-frequency domains where a set of time-annotated input parameters have to be injected into the engine, and several extra conditions (mainly filter and complete-conditions) have to be attached for every skeletal plan for it to run correctly. Even so, the resulting care plan cannot be restricted to available resources, its visualization is very unfriendly for the doctors, and so inappropriate for the goal we pursue: to offer IT support in order to improve the current manual planning stage of the care process.

Consequently, we have used Asbru in order to obtain an intermediate computational representation of the CPG, something very useful for our approach, given the difficulty of directly using traditional P&S languages for modeling a CPG. So, once the Asbru representation of the guideline is translated into an HTN planning domain, and later interpreted by the planner, we obtain the sequencing of the clinical tasks, tailored to patient and available resources. We can observe next the contribution of our methodology applied to our case study, obtaining a personalized patient careflow where the decisions about the treatment are automatically carried out: i) supporting on the CPG control-flow information and the patient profile, ii) respecting the temporal patterns for drug administration, and iii) allocating the resources on the basis of their availability schedule. The output of the planner is similar to the following fragment, for the treatment of patient *Alice*, showing the following information: start and end dates of step, duration of drug administration, iteration, the oncologist in charge (Paul or John), the drug administered, the patient, and the dose.

```
...
22/11/2007; 22/11/2007; Duration:1.0; It14 John AdminPRD; Alice gets dose: 60.0
08/11/2007; 23/11/2007; Duration:360.0; It0 John AdminPRC; Alice gets dose: 100.0
08/11/2007; 09/11/2007; Duration:24.0; It0 John AdminADR; Alice gets dose: 40.0
22/11/2007; 23/11/2007; Duration:24.0; It1 Paul AdminADR; Alice gets dose: 40.0
06/12/2007; 07/12/2007; Duration:24.0; It0 Paul AdminVCR; Alice gets dose: 1.5
...
```

A more user-friendly visualization of the plan can be found online at `http://tratamientos.iactive.es` (just select a profile, e.g. "Maria Casares", a start date, then click the button "Generar Plan", to obtain the plan as a Gantt diagram). The optional translation into a workflow model was described in [8].

Note also that our methodology could be extrapolated to other clinical protocols, since we have shown how to deal with the patterns usually managed in CPG's, i.e. multiple task ordering schemas, delays, synchronizations and cycles. Moreover, it could be adapted to use a different CPG language. Undoubtedly, formalizing the guideline (step 2) will remain a difficult task, since it was manually done, but the protocol we modeled was very well structured and the decisions to be taken on each profile were accurately described in the guideline text, so it was straightforward to use the DELT/A tool for its formalization. Nonetheless, several knowledge acquisition tools were developed to help with this task (e.g. *DeGeL* [10]) that could be integrated in the methodology to facilitate this step.

6 Conclusions

We presented in this paper an AI-based knowledge engineering methodology to develop, model, and operationalize care pathways, thus providing an evidence-based Decision Support System for oncology treatments, and illustrating it with Hodgkin's disease. This is carried out starting from the medical knowledge existing in a previously defined CPG, represented in Asbru. Thus, by means of a translation into an HTN planning domain, and through deliberative reasoning, we achieved a solution for the sequencing and scheduling of the tasks involved in the specific protocol, developing a patient-focused computerized care pathway that respects the patient profile, the available resources and the protocol temporal patterns. As future work, we want to test our framework with a real use case in the hospital (i.e. extracting and integrating the patient profile directly from his/her electronic health record) since the oncologists that participate in the *Oncotheraper* project [6] have validated our results. Further tests with a different disease protocol could be also carried out, in order to check the portability of our method.

References

1. Alexandrou, D., Skitsas, I., Mentzas, G.: A holistic environment for the design and execution of self-adaptive clinical pathways. IEEE Trans. Inf. Technol. Biomed. 15(1), 108–118 (2011)
2. Anselma, L., Montani, S.: Planning: supporting and optimizing clinical guidelines execution. In: Computer-based Medical Guidelines and Protocols: a Primer and Current Trends, pp. 101–120. IOS Press, Amsterdam (2008)
3. Anselma, L., Terenziani, P., Montani, S., Bottrighi, A.: Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. Artificial Intelligence in Medicine 38(2), 171–195 (2006)
4. Castillo, L., Fdez-Olivares, J., García-Pérez, Ó., Palao, F.: Efficiently handling temporal knowledge in an HTN planner. In: 16th ICAPS, pp. 63–72 (2006)
5. Erol, K., Hendler, J., Nau, D.S.: HTN planning: Complexity and expressivity. In: 20th AAAI Conference, pp. 1123–1128. AAAI Press, Menlo Park (1994)
6. Fdez-Olivares, J., Castillo, L., Cózar, J., García Pérez, O.: Supporting clinical processes and decisions by hierarchical planning and scheduling. Computational Intelligence 27(1), 103–122 (2011)
7. Fox, M., Long, D.: PDDL2.1: an extension to PDDL for expressing temporal planning domains. Artificial Intelligence Research 20(1), 61–124 (2003)
8. González-Ferrer, A., Fdez-Olivares, J., Sánchez-Garzón, I., Castillo, L.: Smart Process Management: Automated Generation of Adaptive Cases based on Intelligent Planning Technologies. In: 8th BPM (Demonstration Track), pp. 28–33 (2010)
9. González-Ferrer, A., Fdez-Olivares, J., Castillo, L.: From business process models to hierarchical task network planning domains. Knowl. Eng. Rev. (accepted, 2011)
10. Hatsek, A., Young, O., Shalom, E., Shahar, Y.: DeGeL: a clinical-guidelines library and automated guideline-support tools. In: Computer-Based Medical Guidelines and Protocols: A Primer and Current Trends, pp. 203–212 (2008)
11. Isern, D., Moreno, A.: Computer-based execution of clinical guidelines: a review. International Journal of Medical Informatics 77(12), 787–808 (2008)

12. Kelsey, S.: Managing patient care: are pathways working? Practice Development in Health Care 4(1), 50–55 (2005)
13. Miksch, S., Seyfang, A.: Continual Planning with Time-Oriented, Skeletal Plans. In: Proceedings of ECAI (2000)
14. Miksch, S., Shahar, Y., Johnson, P.: Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans. In: 7th Workshop on Knowledge Engineering: Methods & Languages, pp. 1–25 (1997)
15. Mulyar, N., van der Aalst, W., Peleg, M.: A pattern-based analysis of Clinical Computer-interpretable Guideline modeling languages. J. Am. Med. Inform. Assoc. 14(6), 781–787 (2007)
16. Panella, M., Vanhaecht, K., Sermeus, W.: Care pathways: from clinical pathways to care innovation. International Journal of Care Pathways 13(2), 40–50 (2009)
17. Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R.A., Hall, R., Johnson, P.D., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E.H., Stefanelli, M.: Comparing computer-interpretable guideline models: A case-study approach. J. Am. Med. Inform. Assoc. 10(1), 52–68 (2003)
18. Quaglini, S., Stefanelli, M., Cavallini, A., Micieli, G., Fassino, C., Mossa, C.: Guideline-based careflow systems. Artif. Intell. Med. 20(1), 5–22 (2000)
19. Terenziani, P., German, E., Shahar, Y.: The temporal aspects of clinical guidelines. Studies In Health Technology And Informatics 139, 81–100 (2008)