J Med Syst (2017) 41: 13
DOI 10.1007/s10916-016-0655-6

EDUCATION & TRAINING

Decision Support Provided by a Temporally Oriented Health Care Assistant

An Implementation of Computer-Interpretable Guidelines

Tiago Oliveira[1] · António Silva[1] · José Neves[1] · Paulo Novais[1]

Received: 31 May 2016 / Accepted: 31 October 2016 / Published online: 26 November 2016

Abstract The automatic interpretation of clinical recommendations is a difficult task, even more so when it involves the processing of complex temporal constraints. In order to address this issue, a web-based system is presented herein. Its underlying model provides a comprehensive representation of temporal constraints in Clinical Practice Guidelines. The expressiveness and range of the model are shown through a case study featuring a Clinical Practice Guideline for the diagnosis and management of colon cancer. The proposed model was sufficient to represent the temporal constraints in the guideline, especially those that defined periodic events and placed temporal constraints on the assessment of patient states. The web-based tool acts as a health care assistant to health care professionals, combining the roles of focusing attention and providing patient-specific advice.

This article is part of the Topical Collection on *Education & Training*.

✉ Tiago Oliveira
toliveira@di.uminho.pt

António Silva
pg25307@alunos.uminho.pt

José Neves
jneves@di.uminho.pt

Paulo Novais
pjon@di.uminho.pt

[1] Algoritmi Research Centre/Department of Informatics, University of Minho, Braga, Portugal

Keywords Clinical decision support · Computer-interpretable guidelines · Ontologies · Temporal representation · Reminder systems

Introduction

Although exhaustive proof of the advantages of the widespread use of Clinical Decision Support Systems (CDSSs) [10] is lacking, there is isolated evidence of outcome improvements brought on by these systems in specific settings [1, 6, 8]. Current challenges in CDSS development are mainly concerned with making these systems user-centric and easily accessible by prioritizing and filtering the recommendations that are presented to users at a given time and place [18].

A way to answer the challenges presented to CDSSs is to create functionalities that enable health care professionals to track and follow up their patients, schedule clinical procedures that should be performed, and manage the temporal constraints placed on those procedures. Machine-readable versions of Clinical Practice Guidelines (CPGs) make the answer to these challenges possible. The use of Computer-Interpretable Guidelines (CIGs) [7, 14] endows systems with the capability of providing decision support across different clinical domains and situations, from diagnosis to treatment, determining what questions to ask, tests to perform, the value of results, and paths to follow.

The work presented herein tackles a gap in the development of CIGs, which is the absence of efforts to include a systematic approach for embedding temporal constraints and capabilities in CIG CDSSs [14]. Such features are

important when CIGs are deployed in sensitive contexts, like those involving the prescription of different drug regimens that must be taken at specific times, or the enactment of therapies at particular moments. These are situations that cannot be overlooked during medical practice as their mismanagement may bring drastic consequences upon patients. Therefore, tools to detect errors and monitor their execution actively contribute to the improvement of care. Taking this into consideration, the contributions of this work are twofold:

- The first is a temporal representation model for CIG clinical tasks that allows for the expression of a variety of temporal patterns, including durations, periodicities, task scheduling, time delays and the temporal assessment of patient conditions. To demonstrate the expressiveness of the model, a short example featuring a CPG is provided;
- The second is a form of implementation of CIG temporal execution that uses the developed temporal representation. This is provided as a tool that builds an agenda for the health care professional with the activities that he has to perform. The tool schedules the execution of clinical tasks and keeps track of their execution while trying to promote the fulfilment of their temporal constraints.

The underlying model for CIGs is formalized in Web Ontology Language (OWL) [9], with a particular focus on the temporal representation of clinical tasks, which is crucial for the automatic interpretation of clinical recommendations and their integration in the daily practice of health care professionals. The functionalities of the system are delivered through a web application posing as a health care assistant that provides recommendations for handling a patient, controls their execution times, and provides notifications of their temporal landmarks. In terms of roles, the aim is to develop a tool that focuses the attention of health care professionals and provides patient-specific recommendations.

This article is organized in four sections. Section "Related work" contains a description of the main existing models and tools for the temporal representation and execution of CIGs, their strengths, and their limitations. Section "Development of the health care assistant" presents the architecture of the system and its temporal model for CIG recommendations. It also describes a case study used to demonstrate the expressiveness of the model and the approach followed to make CPGs represented according to it available for execution. Section "Conclusions and future work" presents conclusions about the work developed so far and future work considerations.

Related work

The temporal constraints in CPGs are used to express a variety of elements that need to be controlled in order to ensure the correct application of recommendations and the proper management of patients. Their correct interpretation is vital for the integration of CPG recommendations in the practice of health care professionals. In this regard, it was possible to identify two main groups of temporal constraints [2, 5, 15, 16, 20]. The first group includes temporal constraints about the execution of clinical tasks, which determine when tasks should start and end. The following temporal patterns are featured in this group:

- *Durations*: restrictions that specify for how long a task should be executed;
- *Repetitions*: restrictions that specify how many times a task should be executed;
- *Periodicities*: restrictions that specify how often a task should be executed and the time interval between executions;
- *Waiting Times*: restrictions that specify how long it is necessary to wait between the ending of a previous task and the start of a new task;
- *Repetition Conditions*: restrictions that specify conditions regarding the state of the patient that must hold true before the repetition of a task.

Despite being closely related, the concepts of repetition and periodicity convey different meanings. In the context of CIG representation and execution, a repetition only specifies how many times a task should be executed, with no reference to the time between executions, whereas a periodicity specifies how regularly a task execution should take place. A periodicity is usually bound by a limit to the execution of a task, which may be expressed as a repetition value, the same is to say a task should be performed with a specific frequency, a certain number of times.

The second group encompasses temporal constraints about the state of the patient. They are used to specify the temporal horizon over which a patient will manifest, or should have manifested, a health state. In this sense, they may be used to reason about the past or the future of the patient.

Table 1 shows an assessment of the most prominent models for CIG representation with regards to the above mentioned groups of temporal restrictions. Except for Arden Syntax [15], which provides representation primitives only for one clinical recommendation, all the other models allow the definition of networks of clinical tasks.

Arden Syntax [15] represents clinical recommendations as independent modules, called Medical Logic Modules

Table 1 Assessment of CIG models. The ✓indicates that the model fully represents the temporal constraint in question and the ✗indicates the model does not represent it or has limitations in representing it

CIG Model	Temporal constraints about the execution of tasks					Temporal constraints about the state of a patient
	Durations	Repetitions	Periodicities	Waiting Times	Repetition Conditions	
Arden Syntax [15]	✓	✗	✗	✓	✗	✗
GLIF3 [5]	✓	✗	✗	✗	✗	✓
Asbru [16]	✓	✓	✓	✓	✗	✗
PROforma [20]	✓	✓	✗	✓	✓	✗
GLARE [2]	✓	✓	✓	✓	✓	✗

(MLMs), comprising relevant knowledge for only one decision step. In this sense, a whole CPG cannot be represented in a MLM instance, but rather in a set of isolated MLMs. This is regarded as one of the major limitations of this CIG model. Each MLM is an ACII file divided in three partitions: maintenance, library and knowledge. The actual clinical recommendation is contained in the knowledge partition, where, besides knowledge slots for the definition of actions and rules involving clinical parameters, there are slots for the definition of temporal constraints to be applied to the clinical recommendation. However, there are temporal slots only for durations and waiting times of actions. As shown in Table 1, all of the other above-mentioned temporal constraints are absent from this model, which indicates that Arden Syntax presents important limitations in this regard.

The Guideline Interchange Format (GLIF3) [5] follows a task network model in which a CPG contains different steps, namely: decision steps, patient state steps, branch steps, synchronization steps or action steps. This approach was developed in order to build flowcharts of clinical procedures. In terms of temporal constraints, GLIF3 only allows the definition of durations for actions and decisions, leaving out the definition of repetitions, periodicities, waiting times, and repetition conditions. Despite this, among the selected models of Table 1, it is the only one that allows for the definition of temporal constraints about the state of a patient, on patient data, to evaluate their occurrence. Yet, these temporal constraints can only be defined for the verification of patient states that have occurred retrospectively. It is not possible to define that a certain state should be verified in the future, within a certain time.

A CIG model that offers a comprehensive representation of this pattern is Asbru [16]. This formalism is based on the notion of plan, which represents a CPG. The knowledge required to perform a plan is defined by its knowledge roles, which include preferences, intentions, conditions, effects and plan body. The content of a plan body is composed of other plans until they are no longer decomposable and stand for activities. Asbru temporal annotations allow the definition of durations, with additional specification of starting and ending points of a CIG activity. Furthermore, this can be done with a degree of uncertainty, as these time points can be expressed as intervals. An activity within a plan can also be bounded by repetitions, periodicities and waiting times. As described in Table 1, it is not possible, however, to define repetition conditions or temporal constraints about the state of a patient.

In the PROforma model [20], CPGs are modelled as plans. Each one contains atomic tasks such as actions, decisions and enquiries. The temporal constraints present in this CIG model allow the specification of task duration, waiting times between tasks and repetition conditions of actions. PROforma does not allow for the definition of periodicities and temporal constraints about the state of a patient.

As seen in Table 1 the GuideLine Acquisition, Representation and Execution (GLARE) [2] model is the most complete. GLARE defines a proprietary graph-based structure, where a clinical action is represented by a node. It is possible to define atomic actions like queries to obtain information, work actions that represent medical procedures, decision actions with sets of conditions, and conclusions that describe the output of a decision. Besides the representation elements for the definition of durations, repetitions, waiting times, and repetition conditions, this model is specialized in the representation of periodic actions, offering a wide array of constructs for this temporal pattern. Its only drawback is the absence of temporal constraints about the state of a patient.

It is possible to see that each model has at least one limitation in one type of temporal constraint. From that, one may conclude that the efforts in defining new CIG models

have not been coupled with a simultaneous effort to devise comprehensive temporal constructs in said models.

A crucial component to the operationalization of CIGs is an execution engine that interprets the knowledge formalized in a given model and is capable of making inferences upon it, and a tool to deliver those inferences to health care professionals in the form of recommendations. The last should also present these recommendations to the users and enable inputs to feed the inference process of the execution engine. Examples of such tools include the Guideline Execution Engine (GLEE) [21], SAGEDesktop [4], or the execution engine of GLARE [19]. However, these tools are limited and focus mainly on displaying CPGs as oriented graphs, with no means of integration of the recommendations provided by CPGs in the daily schedule of health care professionals [7].

Development of the health care assistant

The solution proposed for the challenges mentioned in Section "Introduction", i.e, tailoring the recommendations of CDSSs to the context (namely to the temporal dimension), is supported by an architecture such as the one of the CompGuide system, represented in Fig. 1. The architecture brings together all the elements that make possible patient tracking, patient follow-up, scheduling of procedures, and monitoring of procedure constraints based on the temporal elements of CIG clinical tasks. A representation model for temporal constraints plays an important part in the whole work-flow of CIG deployment and execution.

Figure 1 gathers a set of elements aimed at providing timely CPG advice to health care professionals. As a system, it assumes the role of a reminder tool for focusing attention and producing patient-specific advice, by means of its end user application, the Health Care Assistant (HCA).

Section "Architecture of the supporting system" describes each element of the architecture and how they connect to each other. Then, Section "Representation of computer-interpretable guidelines" focuses on the CIG representation model used for the CPGs in the *Guideline Repository*. After this explanation, Section Temporal elements of guidelines explains the temporal elements of the model, which determine how recommendations are interpreted by the execution engine and have influence in the way they are presented to a health care professional. This presentation is dealt with in Section Web-based tool for the visualization and execution of guidelines, which provides a description of the HCA tool for the visualization and execution of CIG recommendations.

Architecture of the supporting system

The architecture of the CompGuide system is shown in Fig. 1. Its main component is a *Core Server* that encapsulates the most important modules of the system. The *Core Server* provides all the required services to allow external applications, such as web applications or mobile applications, to execute guidelines.

The *Authentication Agent* is responsible for authenticating and authorizing the user to access the services of the system and, thus, allowing the access to the functionalities of the *Execution Engine*. It makes distinctions between two types of users, those who can only manipulate information about guideline executions, *simple users*, and those who, in addition, can manipulate information about other users, *admins*.

The required methods to manage and process data about patient profiles, patient states, guideline executions, and tasks to be applied or currently being applied are defined in the *Database Handler*.

The system's knowledge base, i.e., the CPGs encoded in a machine readable format, are in a *Guideline Repository* accessed through a *Guideline Handler* module using the OWL API. This module provides the clinical tasks and respective constraints to the *Guideline Execution Engine* for

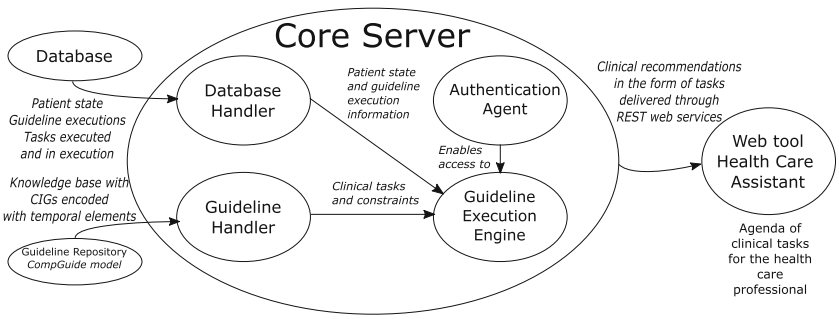

Fig. 1 Architecture of the CompGuide and modules in the *Core Server*

interpretation. The model used for CIG representation is the CompGuide model, based on OWL. As described further ahead, this model incorporates temporal elements for the different temporal patterns that allow for the definition of constraints in clinical tasks.

The *Guideline Execution Engine* performs verifications on task ordering and task constraints by comparing the guideline careflow with the state of the patient. The result is a recommendation in the form of the next clinical task to be applied. The constraints, including temporal constraints, are defined directly in the ontology. Semantic Web Rule Language (SWRL) is not used for this specification due to the flexibility and complexity required for this definition.

The *Core Server* makes the functionalities of the *Guideline Execution Engine* available through a set of RESTful web services for: next task calculation, verification of pending guideline executions, and editing of patient information. The *Core Server* is implemented in Java, using the RESTEasy API over a WildFly Application Server. The notion of CPGs as services, present in CompGuide [13], aims to facilitate the integration of CIGs into any type of application and make them widely accessible, thus enabling differently oriented implementations. The HCA is one such implementation.

Representation of computer-interpretable guidelines

The *Guideline Repository* contains instances of different CPGs in a machine-readable format. The model used is the CompGuide ontology [12], which provides representation primitives for clinical recommendations based on OWL. It follows a task network model in which each recommendation assumes the form of a task. The tasks are connected to each other to build a work-flow of clinical procedures. Although the focus of this paper is placed on the temporal aspects of CIGs, it is important to present the basic structure of a CPG in CompGuide. The key task classes are subclasses of *ClinicalTask*, as can be seen in Fig. 2. They include the following:

- *Action*: a task that should be performed by a health care professional such as an observation, procedure, exam, or treatment application;
- *Question*: a task to get information about the clinical parameters that build the state of the patient;
- *Decision*: a task that encodes a decision regarding the state of a patient;
- *Plan*: a composed task containing instances of the other tasks, defined to achieve a specific goal.

The relative order of clinical tasks in CompGuide is defined with object properties connecting task instances. In this regard, it is possible to define sequential tasks, parallel tasks which should be executed simultaneously, and alternative tasks from which one is automatically selected for execution. In this sense, a guideline in CompGuide resembles a linked list of recommendations. Additionally, it is possible to define different types of conditions that constrain task execution, including trigger conditions to select one amongst alternative tasks, pre-conditions which must be verified before executing a task, and expected outcomes for clinical tasks. The *Condition* class allows the representation of these conditions with specific properties for clinical parameters and their values.

Temporal elements of guidelines

Figure 2 shows a diagram of the classes representing temporal elements in the ontology and their relationship with

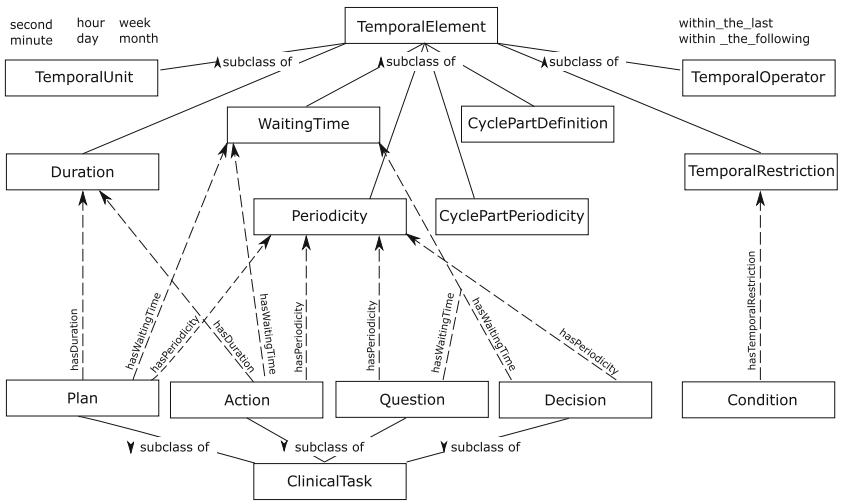

Fig. 2 Classes of the temporal model used in CompGuide

non-temporal classes such as clinical tasks and conditions. The classes that enable the representation of temporal restrictions are all subclasses *TemporalElement* [17]. One of those subclasses is *TemporalUnit* which represents the different units in which a temporal constraint may be expressed. It is an enumerated class consisting of the instances *second*, *minute*, *hour*, *day*, *week*, *month*, and *year*. The remaining classes enable the definition of temporal restrictions about the execution of tasks and temporal constraints about the state of a patient.

Temporal constraints about the execution of clinical tasks

The *Duration* class enables the definition of how long *Actions* and *Plans* should last, since these are the only tasks that may unfold continuously over time. A task instance is connected to a *Duration* instance through the *hasDuration* object property. There are two ways of defining *Duration* instances, as shown in Fig. 3. The first is defining a minimal and maximal duration with the data properties *minDurationValue* and *maxDurationValue*, which contain numerical decimal values. The alternative is to define a fixed duration for the clinical task with the property *exactDurationValue*. Within a *Duration* instance these properties are associated with a *TemporalUnit* through the *hasTemporalUnit* object property, which connects them to one of the above-mentioned instances of the class. Regarding the interpretation of *Duration*, when an *Action* or a *Plan* with this temporal pattern is selected for execution by the *Execution Engine*, the HCA determines its temporal landmarks, i.e., its starting point and ending point(s).

Often times there are instructions in a CPG to delay a procedure in order to observe the evolution of a patient. In the CompGuide ontology this is expressed with an instance of the *WaitingTime* class, by connecting the clinical task that should be delayed to the instance through the *hasWaitingTime* object property. These delays can be defined for any type of task. In Fig. 4, it is shown that the *minWaitingTimeValue* and *maxWaitingTimeValue* data properties are used when one aims to express the earliest and latest possible starting points of the task, after a previous task is finished. If the delay is a fixed value, then it is expressed with the *exactWaitingTimeValue*. The *hasTemporalUnit* property is used again to specify the units. The temporal landmarks produced by the HCA upon the interpretation of this task consist of its possible starting points.

A periodic task is defined using the property *hasPeriodicity*, which connects the task to an instance of the class *Periodicity*. This temporal pattern can be defined for any type of task. As shown in Fig. 5, an instance of *Periodicity* can also be connected to an instance of *Duration* through the *hasDuration* object property, thus determining for how long a periodic task should take place. If one wants to state the number of times the event should take place, i.e., the number of cycles of the periodic task), it is necessary to formulate a repetition constraint, which is possible with the *repetitionValue* data property, with a range of integer numerical values. It could also be the case the periodic task should only occur until a condition about the state of a patient is verified. To express this, one uses the *hasStopCondition* object property to connect an instance of *Periodicity* to instances of the class *Condition*. While it is possible for a periodicity

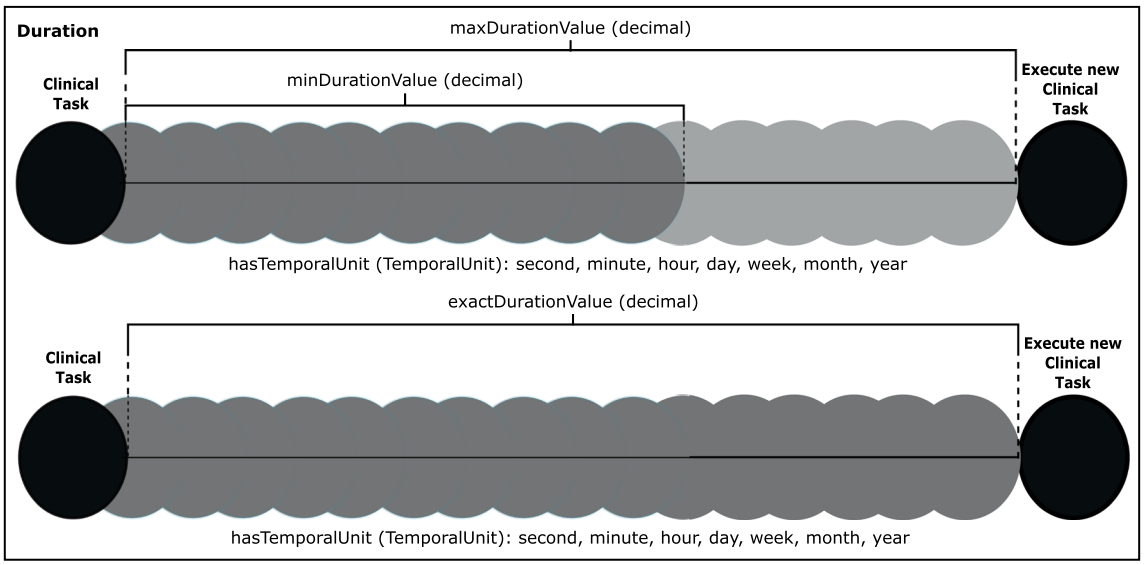

Fig. 3 Representation of a *Duration* applied to a clinical task

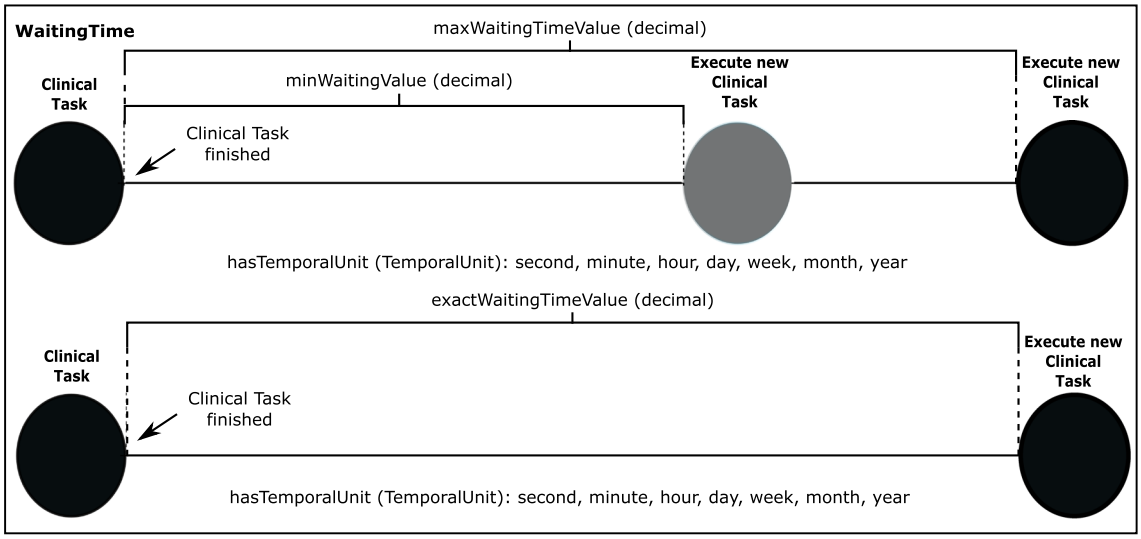

Fig. 4 Representation of a *WaitingTime* applied to a clinical task

to have a duration and a stop condition, a repetition value and a stop condition, or just a stop condition, it is not possible to have both a duration and a repetition value because it is considered to be redundant information. The stop condition takes precedence over the other temporal restrictions, and, if the condition is met, the task is immediately stopped. The frequency of the event is defined in the data property *periodicityValue* and the associated *TemporalUnit*. A periodic task is thus unfolded in a series of executions handled as events. In turn, each event may have an associated periodicity or duration. These nested temporal patterns are defined with the *hasCyclePartDefinition* object property, connecting the *Periodicity* instance to a *CyclePartDefinition* instance, within which it is possible to define a duration with the reuse of the *Duration* class or a new periodicity with the *CyclePartPeriodicity* class. The temporal landmarks produced by the HCA for these temporal constraints consist of every execution, with starting and ending points, of the events of the periodic task.

Temporal constraints about the state of a patient

Temporal reasoning about the state of a patient is enabled by the *TemporalRestriction* class, whose instances can be associated with a *Condition* through the *hasTemporalRestriction* property. With the *hasTemporalOperator* property a *TemporalOperator* is specified for the restriction.

Being an enumerated class, *TemporalOperator* consists of two instances, *within_the_last* and *within_the_following*. The operator *within_the_last* is used when one wants to express that a condition about the patient state must have held true at least once, within a period of time just before execution time. It is used in trigger conditions, preconditions and conditions of rules in *Decision* instances.

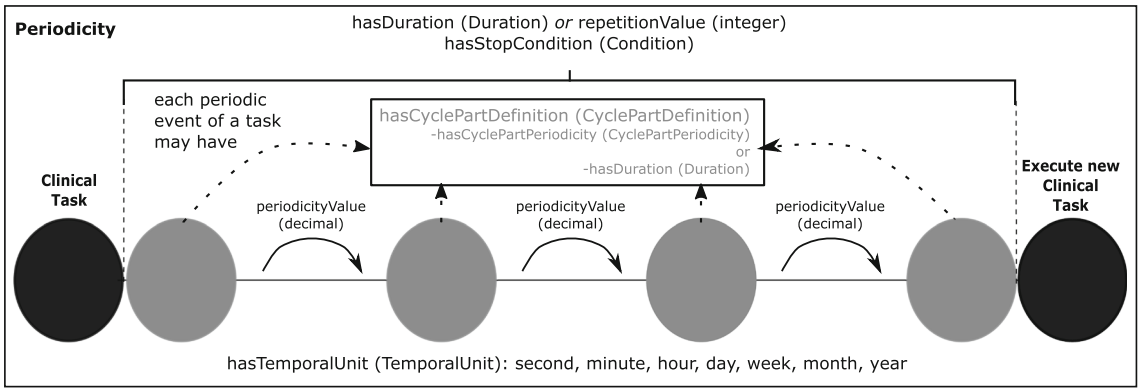

Fig. 5 Representation of a *Periodicity* applied to a clinical task

The execution engine interprets this operator by checking if, in the state of the patient, there is a record regarding the parameter in the condition, registered within the specified time frame, whose value validates the condition. As for the *within_the_following* operator, it expresses a condition about the future, in which one aims to observe the effect a clinical task has after being applied to a patient. Such conditions are used in task outcomes.

The temporal operators represent the reach of a temporal constraint and are used together with temporal units, defined through the *hasTemporalUnit* object property. The temporal restriction values are expressed through data properties such as *maxTemporalRestrictionValue* and *minTemporalRestrictionValue* for an interval, or *temporalRestrictionValue* for an exact value, with a range of decimal numerical values.

Case-study featuring a guideline for colon cancer treatment

In order to verify the expressiveness of the temporal representation model, a guideline of the National Comprehensive Cancer Network (NCCN) [3] was fully represented using the CompGuide ontology. The CPG is used for the diagnosis and management of colon cancer and, thus, contains many and varied clinical tasks with temporal constraints. The process of representing the guideline was accomplished using Protégé and resulted in an ontology *owl* file with 223 task instances, of which a large majority (190) consisted of *Action* tasks. Among the clinical tasks, 95 of them had temporal constraints. The most common type of temporal constraint was the *Periodicity*, featured in 79 tasks, most of them limited by a duration. There were also 7 tasks with nested periodicities using *CyclePartDefinition*. The reason for such an abundance of periodicities is the detailed descriptions of chemotherapy regimens in the CPG. The remaining temporal restriction cases were 7 instances of *Duration* and 2 instances of *WaitingTime*. The temporal classes and their respective properties enabled the representation of all the temporal patterns in the CPG. Figures 6 and 7 show the instantiation of case examples for each temporal pattern.

As can be seen in the figures, the duration of Case 1 is expressed with an interval and the waiting time of Case 2 is expressed with an exact value. The interpretation of the HCA in Case 1 would be to establish in the calendar of the health care professional (the user) the starting and ending times of neoadjuvant therapy, notifying him of when the task should start and when it reaches its earliest ending time and latest ending time. As for Case 2, the HCA would not let the reevaluation start right after the ending of chemotherapy and would notify the health care professional of when the task should start, i.e., after 2 months.

Cases 3, 4, and 5 from Fig. 6 represent situations of *Periodicity*. In Case 3 the periodicity of the physical exam is bounded by a duration, which means that the HCA would tell the user every 6 months during 2 years that he should perform the exam. This represents the unfolding of a clinical task into multiple occurrences to which we call events. The difference to Case 4 is that the last is also bounded by a stop condition. Upon the ending of each colonoscopy event, the HCA would ask the health care professional if signs of adenoma were found and, if that were the case, it would finish the task and recommend the next procedure. Case 5 has a nested periodicity that is interpreted by the HCA in the following way: besides notifying the user every 3 months of the chemotherapy, within each event, it would alert the user to the administration of chemotherapy substances every 12 hours during 14 days. The 3 months to the next chemotherapy event would start counting again after those 14 days.

In the representation of the CPG there were only 6 occurrences of temporal constraints about he state of a patient. Case 6 represents the typical situation of expressing the outcome of a chemotherapy task. In this case, the HCA, following 6 months from the end of chemotherapy, would ask the user if the tumor became operable and the objective of the task was fulfilled. Depending on the answer, different procedures would be selected according to the CPG careflow. Thus, it is a condition defined for the future of the patient. As for Case 7, it configures a situation of a trigger condition for the selection of a chemotherapy regimen which has an associated temporal restriction, in order to avoid conflicts with different chemotherapy regimens. In such a case, the HCA verifies if there is a regorafenib chemotherapy regimen in the patient record within 12 months prior to execution time. Only if that were the case, would the experimental chemotherapy be selected.

In terms of expressiveness, this approach goes beyond models like Arden Syntax and GLIF3 since it provides a wider set of different temporal constraints. Taking into account what is presented in Section "Related work", Arden Syntax enables the definition of durations for decisions, which the presented temporal elements of CompGuide do not. However, this was a design decision as it was considered that only actions can unfold over time and, thus, have a duration. The possibility of defining temporal constraints about the state of the patient and temporally reason about patient data is the major strength of GLIF3. The CompGuide model also possesses such temporal constraints, with the additional feature of defining temporal constraints for the outcomes of actions, and not only retrospectively. Moving on to the representation of repetitions, periodicities and waiting times, the CompGuide temporal elements are on par with Asbru, PROforma and GLARE. The treatment

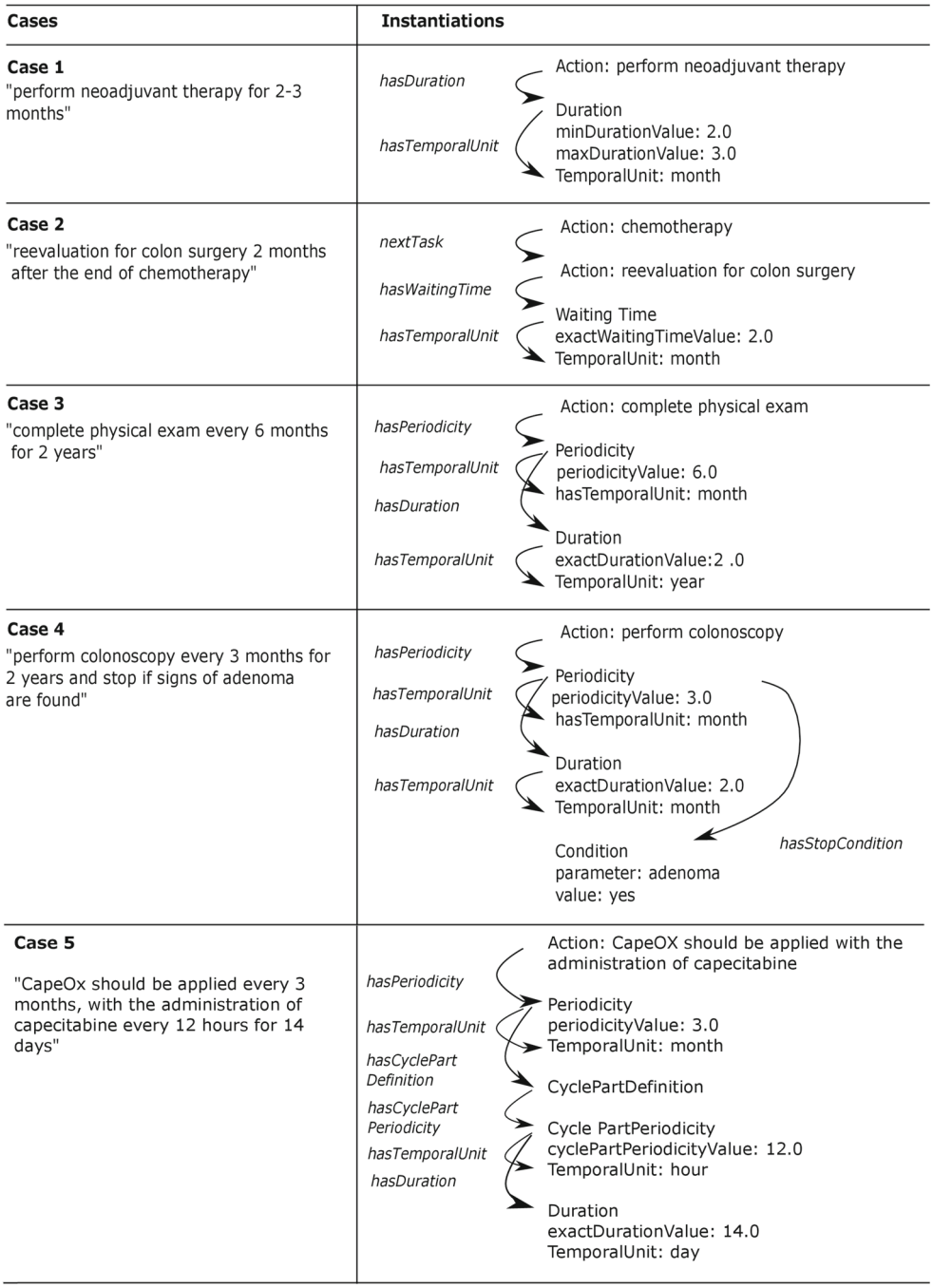

Fig. 6 Instantiation of case examples for *Duration*, *WaitingTime* and *Periodicity*

of periodicities is the most important aspect in view of the frequency with which this temporal pattern appears in CPGs. In this regard, the handling of nested periodicities is similar to the one performed in GLARE. An important difference in interpretation is that of repetition conditions. In the proposed model, rather than defining a condition for the continuous execution of a task (as in GLARE), the choice was to define a condition which, when true, stops the execution of a task. Nonetheless, the practical effects of this are the same. The distinctive feature of the CompGuide temporal elements compared to GLARE is the definition of temporal constraints about the state of a patient.

The objective of the temporal elements described in Section "Temporal elements of guidelines" and exemplified in the current section is to provide an encompassing representation of temporal patterns, without the need for proficiency in a specific programming language for their definition. For this reason, the basic structure of these elements is defined in OWL, but all the meaning behind them and the logics of their interpretation are encoded in the

Fig. 7 Instantiation of case examples for temporal constraints about the state of the patient

Cases	Instantiations
Case 6 "the tumor should become operable after 6 months of FOLFOX or CapeOx chemotherapy"	Action: chemotherapy with FOLFOX or CapeOX *hasOutcome* Condition parameter: tumor status value: operable *hasTemporal Restriction* TemporalRestriction temporalRestrictionValue: 6.0 *hasTemporalUnit* TemporalUnit: month *hasTemporal Operator* TemporalOperator: within_the_following
Case 7 "for therapy after third progression consider experimental chemotherapy, if the regorafenib regimen has been applied within the last 12 months"	Action:experimental chemotherapy after first progression *hasTrigger Condition* Condition parameter: applied regorafenib chemotherapy value: yes *hasTemporal Restriction* TemporalRestriction temporalRestrictionValue: 12.0 *hasTemporalUnit* TemporalUnit: month *hasTemporal Operator* TemporalOperator: within_the_last

Guideline Execution Engine. The execution engine is tailored to the model and, as a result, the two should coexist in a system. The implementation of the temporal model in another CDSS has to be coupled with the *Guideline Execution Engine*. This is one of the reasons for the creation of the *Core Server* of Fig. 1. It exposes the functionalities of

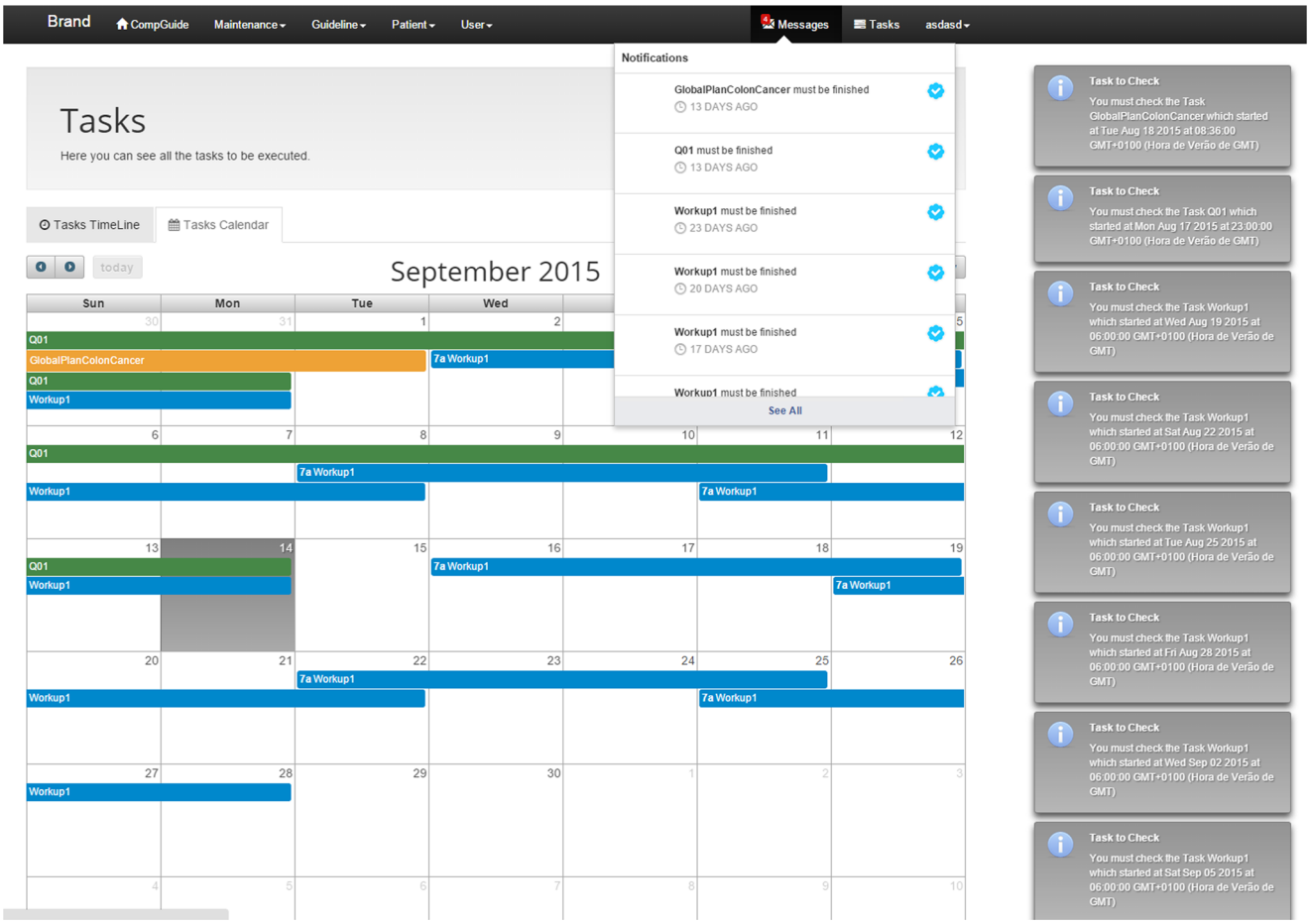

Fig. 8 Calendar task view and notifications of the Health Care Assistant

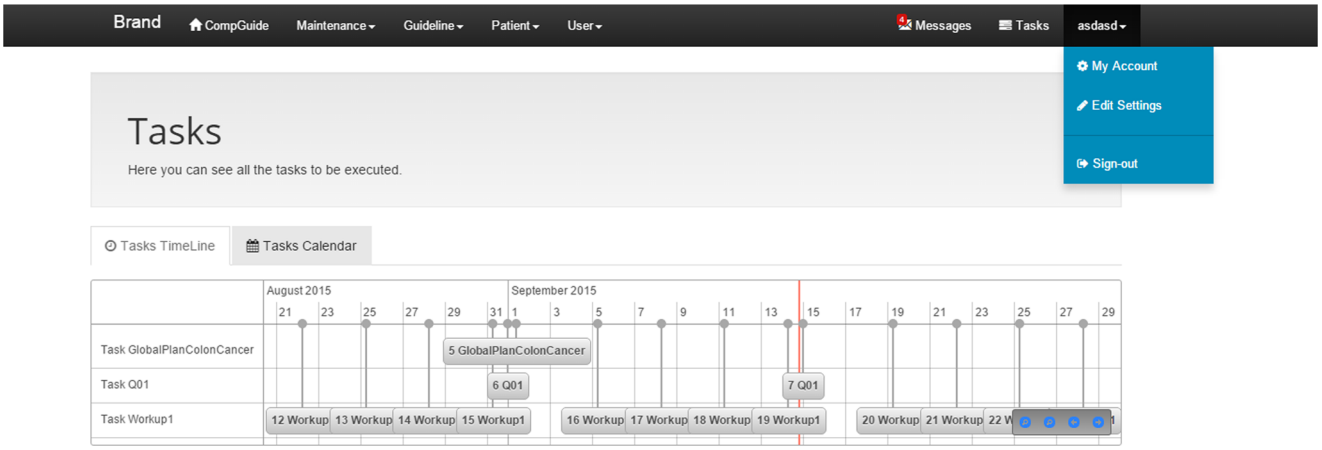

Fig. 9 Timeline task view of the Health Care Assistant

the *Guideline Execution Engine* through a set of web services for integration in external software tools. Provided that the intended CPG is encoded according to the CompGuide model in the *Guideline Repository*, its execution in an external CDSS is possible.

Web-based tool for the visualization and execution of guidelines

The way in which clinical recommendations are delivered to health care professionals may dictate de adoption of a tool for clinical decision support. The temporal elements in the CompGuide ontology enable not only the temporal execution of CPGs but also the development of new ways to visualize CPG advice. Ways that allow health care professionals to accurately track their activities while benefiting from automatic reasoning features, according to the clinical constraints defined in a CIG. The *Guideline Execution Engine* interprets the content of a CIG and the generated information is passed on to the HCA, which is the tool through which the health care professional interacts with the advice.

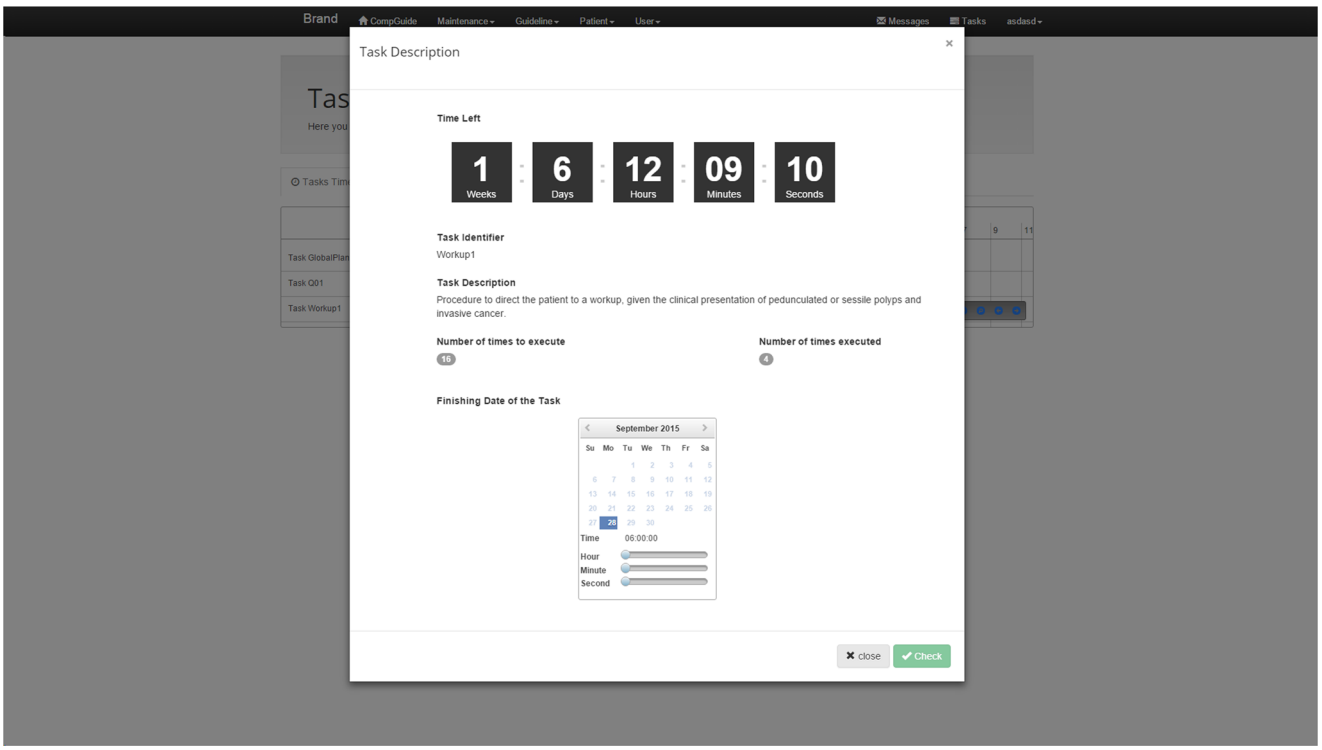

Fig. 10 Description of a clinical task in the Health Care Assistant

The HCA was developed as a web application so that it can be widely available, whichever the platform it is accessed from. Its main objectives are to provide timely clinical recommendations and integrate them in the clinical practice of the health care professional. To fulfil this, it implements the functionalities available in the *Execution Engine*. It was developed following the Model-View-Control (MVC) paradigm using Java Server Faces (JSF).

Besides the automatic calculation of the proper clinical tasks to apply and the validation of conditions regarding the state of the patient placed upon them, based on user inputs, its strength lies in its temporal features. The tool builds a schedule for the health care professional based on the tasks recommended by the *Execution Engine* and their respective temporal constraints, which can be viewed as calendar, as in Fig. 8, or as a timeline, as in Fig. 9. The application informs the users of when they should execute clinical tasks, when they should start them, when they should finish them, and assesses results of expected outcomes.

These two views offer different possibilities to the user, namely the possibility to get an overall view of the clinical process with the calendar view and to focus on a task at a time with the timeline view. The notifications mentioned throughout Section "Case-study featuring a guideline for colon cancer treatment" can be seen as side messages, as shown in Fig. 8. By clicking on a task entry, it is possible to visualize task details such as remaining execution time and number of executions, task descriptions and so forth, as seen in Fig. 10.

Conclusions and future work

The work presented herein is an implementation example of the notion of guidelines as services, presented in [11], which takes advantage of the flexibility of the CompGuide system. The main contributions are a comprehensive temporal representation model and a web-based tool for the execution of CIGs. The tool builds an agenda of clinical tasks for the health care professional to follow and provides timely notifications of clinical events, while filtering the advice given to the health care professionals at a given time. The intention is to lessen the burden placed on them and help to keep their patients on the right track. Compared to current applications for the execution of CIGs, the one presented herein reflects a different view of guideline application and is endowed with functionalities that go beyond the simple display of clinical tasks. It is also a reminder system that may help the user to manage time and to ensure the enactment of procedures. Although there is a functional prototype, the tool is still in development. However, it represents a path to make CPGs more dynamic and improve their daily application.

As future work, it is necessary to evaluate the HCA tool by performing usability tests with health care professionals in order to infer about the usefulness of the developed application. A functionality that is currently being developed is the integration of clinical tasks with the calendar service used by the health care professional, thus enabling the visualization of CIG executions not only on the HCA, but also on their own calendar services, such as Google Calendar or Microsoft Calendar), with the other events of their daily practice which are outside the scope of guideline execution.

Acknowledgments This work has been supported by COMPETE: POCI-01-0145-FEDER-0070 43 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope UID/CEC/ 00319/2013. The work of Tiago Oliveira is supported by a FCT grant with the reference SFRH/BD/85291/ 2012.

References

1. Adhikari NKJ, Beyene J, Sam J, and Haynes RB, Effects of Computerized Clinical Decision Support Systems on Practitioner Performance. *Journal of the American Medical Association* 293(10):1223–1238, 2005.
2. Anselma L, Terenziani P, Montani S, and Bottrighi A, Towards a Comprehensive Treatment of Repetitions, Periodicity and Temporal Constraints in Clinical Guidelines. *Artificial Intelligence in Medicine* 38(2):171–195, 2006. doi:10.1016/j.artmed.2006.03.007.
3. Benson A, Bekaii-Saab T, Chan E, Chen YJ, Choti M, Cooper H, and Engstrom P: NCCN Clinical Practice Guideline in Oncology Colon Cancer. Tech. rep., National Comprehensive Cancer Network, http://www.nccn.org/professionals/physician_gls/f_guidelines.asp, 2013.
4. Berg D, Ram P, Glasgow J, and Castro J, SAGEDesktop: An Environment for Testing Clinical Practice Guidelines. *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 4:3217–3220, 2004. doi:10.1109/IEMBS.2004.1403906.
5. Boxwala Aa, Peleg M, Tu S, Ogunyemi O, Zeng QT, Wang D, Patel VL, Greenes Ra, and Shortliffe EH, GLIF3: A Representation Format for Sharable Computer-Interpretable Clinical Practice Guidelines. *Journal of Biomedical Informatics* 37(3):147–61, 2004. doi:10.1016/j.jbi.2004.04.002.
6. Bright T, and Wong A, Effect of Clinical Decision-Support Systems: A Systematic Review. *Annals of Internal Medicine* 157(1):29–43, 2012. doi:10.7326/0003-4819-157-1-201207030-00450.
7. Isern D, and Moreno A, Computer-based Execution of Clinical Guidelines: a Review. *International Journal of Medical Informatics* 77(12):787–808, 2008. doi:10.1016/j.ijmedinf.2008.05.010.
8. Kaushal R, Shojania KG, and Bates DW, Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review. *Archives of internal medicine* 163(12):1409–16, 2003. doi:10.1001/archinte.163.12.1409.
9. McGuinness DL, and Van Harmelen F: OWL Web Ontology Language Overview. https://www.w3.org/TR/owl-features/, 2004.
10. Musen MA, Shahar Y, and Shortliffe EH, Clinical decision-support systems. In: Shortliffe E, and Cimino J (Eds.) Biomedical

Informatics, Health Informatics, pp. 698–736. New York: Springer, 2006. doi:10.1007/0-387-36278-920.
11. Novais P, Oliveira T, and Neves J, Moving towards a new paradigm of creation, dissemination, and application of computer-interpretable medical knowledge. *Progress in Artificial Intelligence*, 1–7, 2016. doi:10.1007/s13748-016-0084-2.
12. Oliveira T, Novais P, and Neves J, Representation of Clinical Practice Guideline Components in OWL. In: Trends in Practical Applications of Agents and Multiagent Systems SE - 10, Advances in Intelligent Systems and Computing, Vol. 221, pp. 77–85: Springer International Publishing, 2013.
13. Oliveira T, Leão P, Novais P, and Neves J, Webifying the Computerized Execution of Clinical Practice Guidelines. In: Bajo Perez J, Corchado Rodríguez JM, Mathieu P, Campbell A, Ortega A, Adam E, Navarro EM, Ahrndt S, Moreno MN, and Julián V (Eds.) Trends in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection SE - 18, Advances in Intelligent Systems and Computing, Vol. 293, pp. 149–156: Springer International Publishing, 2014. doi:10.1007/978-3-319-07476-4_18.
14. Peleg M, Computer-interpretable Clinical Guidelines: A Methodological Review. *Journal of Biomedical Informatics* 46(4):744–63, 2013. doi:10.1016/j.jbi.2013.06.009.
15. Samwald M, Fehre K, de Bruin J, and Adlassnig KP, The Arden Syntax standard for clinical decision support: Experiences and directions. *Journal of biomedical informatics*, 2012. doi:10.1016/j.jbi.2012.02.001.
16. Shahar Y, Miksch S, and Johnson P, The Asgaard Project: A Task-specific Framework for the Application and Critiquing of Time-oriented Clinical Guidelines. *Artificial intelligence in Medicine* 14(1-2):29–51, 1998. doi:10.1016/S0933-3657(98)00015-3.
17. Silva A, Oliveira T, Novais P, and Neves J, Representing Temporal Patterns in Computer-Interpretable Clinical Guidelines. In: Schulz C, and Liew D (Eds.) Imperial College Computing Student Workshop (ICCSW 2015), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, OpenAccess Series in Informatics (OASIcs), Vol. 49, pp. 62–69, 2015. doi:10.4230/OASIcs.ICCSW.2015.62.
18. Sittig DF, Wright A, Osheroff Ja, Middleton B, Teich JM, Ash JS, Campbell E, and Bates DW, Grand challenges in clinical decision support. *Journal of biomedical informatics* 41(2):387–92, 2008. doi:10.1016/j.jbi.2007.09.003.
19. Terenziani P, Montani S, Bottrighi A, Torchio M, Molino G, and Correndo G, The GLARE Approach to Clinical Guidelines: Main Features. *Studies in Health Technology and Informatics* 101(3):162–6, 2004. doi:10.3233/978-1-60750-944-8-162.
20. Vollebregt A, ten Teije A, van Harmelen F, van der Lei J, and Mosseveld M, A study of PROforma, a development methodology for clinical procedures. *Artificial Intelligence in Medicine* 17(2):195–221, 1999. doi:10.1016/S0933-3657(99)00016-0.
21. Wang D, Peleg M, Tu SW, Boxwala AA, Ogunyemi O, Zeng Q, Greenes RA, Patel VL, and Shortliffe EH, Design and implementation of the GLIF3 guideline execution engine. *Journal of Biomedical Informatics* 37(5):305–318, 2004. doi:10.1016/j.jbi.2004.06.002.