Temporal detection and analysis of guideline interactions

Luca Anselma[a], Luca Piovesan[b*], Paolo Terenziani[b]

[a]Dipartimento di Informatica, Università degli Studi di Torino, Corso Svizzera 185, 10149 Torino, Italy
anselma@di.unito.it
[b]Computer Science Institute, Dipartimento di Scienze e Innovazione Tecnologica, Università del Piemonte Orientale, Viale Teresa Michel 11, 15121 Alessandria, Italy
{paolo.terenziani|luca.piovesan}@uniupo.it

Abstract.

Background: Clinical practice guidelines (CPGs) are assuming a major role in the medical area, to grant the quality of medical assistance, supporting physicians with evidence-based information of interventions in the treatment of single pathologies. The treatment of patients affected by multiple diseases (*comorbid* patients) is one of the main challenges for the modern healthcare. It requires the development of new methodologies, supporting physicians in the treatment of interactions between CPGs. Several approaches have started to face such a challenging problem. However, they suffer from a substantial limitation: they do not take into account the temporal dimension. Indeed, practically speaking, interactions occur *in time*. For instance, the effects of two actions taken from different guidelines may potentially conflict, but practical conflicts happen only if the times of execution of such actions are such that their effects overlap in time.

Objectives: We aim at devising a methodology to detect and analyse interactions between CPGs that considers the temporal dimension.

Methods: In this paper, we first extend our previous ontological model to deal with the fact that actions, goals, effects and interactions occur in time, and to model both qualitative and quantitative temporal constraints between them. Then, we identify different application scenarios, and, for each of them, we propose different types of facilities for user physicians, useful to support the temporal detection of interactions.

Results: We provide a modular approach in which different Artificial Intelligence temporal reasoning techniques, based on temporal constraint propagation, are widely exploited to provide users with such facilities. We applied our methodology to two cases of comorbidities, using simplified versions of CPGs.

Conclusion: We propose an innovative approach to the *detection and analysis of interactions between CPGs* considering different sources of temporal information (CPGs, ontological knowledge and execution logs), which is the first one in the literature that takes into account the temporal issues, and accounts for different application scenarios.

Keywords. Temporal reasoning, Ontology of time and interactions, Medical knowledge representation, Computer-interpretable clinical guidelines, Guideline interaction detection, Comorbidity treatment.

1 Introduction

Clinical Practice Guidelines (CPGs) are – in the definition of the American Institute of Medicine – "systematically developed statements to assist practitioner and patient decisions about appropriate health care in specific clinical circumstances" [1]. Thousands of CPGs have been devised in the last years. For instance, the Guideline International Network [2] comprises 100 organizations representing 48 countries and provides a library of more than 6500 CPGs. The adoption of computerized approaches to acquire, represent, execute and reason with CPGs can further increase

---

the advantages of CPGs. Thus, in the last twenty years, several different approaches to Computer-Interpretable Guidelines (CIGs) have been developed (consider, e.g., [3–6]).

By definition, clinical guidelines address specific clinical circumstances (i.e., specific diseases). Unfortunately, in many cases patients are affected by more than one disease. The treatment of *comorbid patients* (i.e., patients affected by multiple diseases) is one of the main challenges for the modern health care, also due to the aging of population and the consequent increase of chronic diseases. Thus, "*health care systems must deal with an increasing number of patients with several simultaneous pathologies (i.e., comorbid patients). [...] Clinical practice guidelines provide evidence-based information of interventions, but only on individual pathologies. This sets up the urgent need of developing ways of merging multiple single-disease interventions to provide professionals' assistance to comorbid patients.*" [7].

However, though some CPGs covering frequently occurring comorbidities might be devised, the approach of considering all the possible combinations of pathologies does not scale up:

"*Developing Clinical Practice Guidelines that explicitly address all potential comorbid diseases is not only difficult, but also impractical, and there is a need for formal methods that would allow combining several disease-specific clinical practice guidelines in order to customize them to a patient*" [8].

Thus, there is a need for formal methodologies to support physicians in the detection and resolution of interactions between guidelines, and, ultimately, in the process of merging two or more guidelines. This is an increasingly "hot topic" within the Medical Informatics community, and several approaches have been proposed in the last years, including our approach for user-driven and ontology-based detection of relevant interactions at multiple levels of abstraction [9].

Nonetheless, to the best of our knowledge, until now no CIG approach in the literature has taken into consideration the ***temporal dimension*** in the study of interactions. This is, in our opinion, a crucial restriction, which severely limits the practical impact of current approaches. Indeed, a non-temporal analysis can only detect *theoretically possible* interactions between actions in different CIGs, identifying, e.g., a potential conflict between their goals or effects. However, as long as no temporal analysis is performed, such an interaction is only "*potential*": indeed, *actual* interactions occur *in time*, i.e., just in case that the considered goals or effects *overlap in time*. The approach in this paper (a preliminary short version of it has been published in [10]) is, to the best of our knowledge, the first one starting to face such a challenging problem.

To give an idea of the problem, here we briefly describe an example that will be explained in deep as a running example along the paper.

Example 1 (Description). The CIG for deep venous thrombosis prescribes, among the other actions, a treatment with Warfarin. Warfarin must be administered once a day thus, for instance, in the CIG there is a constraint between the first and the second Warfarin administrations binding the delay between them to be greater than or equal to one day and to be less than two days. Warfarin has an anticoagulant effect that appears approximately within a day after the administration, and lasts from a minimum of day to a maximum of four days. When executing the CIG for venous thrombosis on a specific patient, one must consider also the fact that the first two Warfarin administrations have been already performed on days 01/04/2015 and 01/04/2015. In case the patient treated for deep venous thrombosis develops a respiratory tract infection, another CIG must be executed to treat the onset disease. The CIG for the respiratory tract infection suggests, among the others, the treatment with Erythromycin, an antibiotic. The drug interaction between Erythromycin and Warfarin is well known in medical literature (see e.g., [11]) and it is caused by the interaction between the anticoagulant effect of Warfarin and the reduction of metabolism, an effect induced by the Erythromycin. Their co-occurrence of the two effects increases the anticoagulant effect of Warfarin, raising the risk of bleedings. A physician could be interested in answering questions such as:

1) Given the times in which the different actions have been executed, is it possible to have an overlap between the anticoagulant and the reduction of metabolism (i.e., to have interaction)?

2) Does exist a temporal configuration for the future actions that complies with all the constraints imposed by the CIGs that avoid the interactions?

We aim at proposing a general and wide-purpose approach to ***temporal detection of interactions between guidelines***. In our approach, we support three different sources of temporal information:

(i)   The CIGs, which contain temporal constraints between their actions. In Example 1, for instance, the constraints between each pair of consecutive Warfarin administrations. This knowledge is specific to the guideline for the treatment of venous thrombosis.

(ii) The general medical knowledge, which states the temporal constraints between the clinical actions and their effects. In Example 1, for instance, knowledge that Warfarin has an anticoagulant effect that appears approximately within a day after the administration, and lasts from a minimum of day to a maximum of four days. Notice that such a knowledge is relative to the effects of Warfarin in general, independently of the specific guideline in which Warfarin is used. For such a reason we cope with such types of knowledge once and for all, as a guideline-independent piece of knowledge.

(iii) The *logs*, which store the execution times of the CIG actions executed on the specific patients. In Example 1, for instance, the fact that the first two Warfarin administrations have been executed on days 01/04/2015 and 01/04/2015. This source is optional (see below).

In the GLARE (Guideline Acquisition, Representation and Execution) approach [12,13], which is at the basis of our work, CIGs strictly contain the evidence-based knowledge concerning the diagnostic and therapeutic treatment of a disease. However, such a disease-specific knowledge, which is mostly procedural knowledge, is complemented by what we call "general medical knowledge", which is the basic medical knowledge, mostly declarative, which is not related to a specific disease (so that it would be not "economical" to repeat it in many different guidelines). It is worth noticing that the distinction between CIGs and general medical knowledge, already adopted in [10], is not strict. Indeed, other approaches such as ASBRU [14] represent actions effects within CIGs.

For the sake of generality, we have identified four different possible application scenarios for our approach, depending on the type of the analysis (*abstract* or *patient-dependent*) and on the availability and nature of the execution log (*non-temporal*, *temporally imprecise* or *temporally exact*).

For all of the different scenarios, we provide user-physicians with a wide range of facilities, considering the possibility of asking (i) whether interactions between selected actions are temporally possible or necessary, or (ii) the times when actions in CIGs have to be executed in order to avoid/enforce interactions. In addition, *hypothetical* queries are supported, to allow user-physicians to perform "*what-if*" temporal analysis of interactions. Thus, in this paper, we provide the *first* approach that widely tackles the issue of the *temporal* detection of interactions between CIGs, covering different tasks (e.g., patient-dependent vs. patient-independent analysis) and application scenarios and providing a wide range of user-friendly facilities. From the methodological point of view, we have achieved such a challenging task by:

1. Proposing an extended temporal ontology, modelling actions, effects and interactions, as well as a domain-independent formalism to model temporal constraints.

2. Representing temporal constraints (i) in CIGs, (ii) in logs, and (iii) in a knowledge base considering actions, effects and interactions ((iii) is called *ontological knowledge* henceforth).

3. Providing algorithms to extract temporal information from CIGs, logs and ontological knowledge, to merge it and reason with it, and to support the different query answering facilities.

In particular, temporal reasoning and the different facilities are based on the well-known *Simple Temporal Problem* (henceforth STP) framework, which supports *correct* and *complete* temporal constraint propagation on *bounds on differences* constraints in a time *cubic* in the number of events [15]. Notably, however, such a choice is transparent to user-physicians, who can use the high-level temporal formalism described by the ontology to interact with the system.

It is also worth stressing that, although we are implementing our approach on top of the GLARE system, the methodology we propose is largely system-independent.

The paper is organized as follows. Section 2 describes related work, with specific attention to GLARE, highlighting the advances we aim to achieve in our approach. In Section 3, we extend GLARE knowledge representation model to cope also with the temporal aspects, including the definition of a formalism covering different types of qualitative and quantitative temporal constraints. In Section 4, we describe the general architecture for our temporal approach and the

temporal facilities it provides to users. In Section 5, we face the technical temporal reasoning problems, describing in detail the temporal reasoning engine we provide to support the above facilities. Section 6 briefly addresses the issue of providing a suitable visualization of the output of temporal reasoning to user-physicians. Finally, Section 7 contains discussions and conclusions. Additionally, Appendix 1 introduces some basic notions of temporal reasoning and STP. We strongly recommend readers non-specialist in temporal reasoning to read it before the following sections, since it provides some background necessary to understand our contribution. Finally, Appendix 2 contains some tables, representing the output of temporal reasoning for the examples in the paper.

2 Related work

The treatment of comorbid patients is one of the main challenges for the modern healthcare. This is a "hot topic" in Medical Informatics, too, and several approaches are recently emerging to cope with the integration of CIGs to manage comorbid patients. In the following, we first discuss some of the main approaches in the literature (Section 2.1), and then we focus on GLARE and its current extensions coping with comorbidities (Section 2.2).

2.1 CIGs and comorbidities: state of the art

Several approaches to the treatment of comorbidities with CIGs have been recently proposed. Among them, the approach in [16] is the most similar to ours. It provides a conceptual model for medical actions and detects interactions using such a model and first order logic (FOL) rules. However, it does not consider time. Other approaches, for instance [8] and [17], use constraint logic programming to identify and address adverse interactions between actions. In this solution, a constraint logic programming (CLP) model is derived from the combination of logical models that represent the CIGs, then a mitigation algorithm is applied to detect and mitigate interactions. In these approaches, interactions and their managements are defined a priori by physicians. Sánchez-Garzón et al. [18] propose an agent-based approach to guideline merging. Each guideline is considered as a physician expert in the treatment of a single disease and it is represented by an agent with hierarchical planning capabilities. The result is obtained through the coordination of all the agents and respects the recommendations of each guideline. Riaño et al. represent guidelines as sets of clinical actions that are modelled into an ontology [19]. To combine two treatments, first they are unified in a unique treatment and then a set of "combination rules" is applied to detect and avoid possible interactions. A model-based automatic merge of CIGs is then proposed in [7] through the definition of a combining operator. Jafarpour and Abidi [20] use semantic-web rules and an ontology for the merging criteria. Given these, an Execution Engine dynamically merges several CIGs according to merge criteria. Two recent works [21,22] are based on answer set programming (ASP) approaches. Basically, they transform CIGs, patient's information and a set of mitigation operators into an ASP program. Each solution of the ASP program corresponds to a possible execution of the merged CIGs. Among them, the approach in [21] is of particular interest, since it introduces the possibility of expressing preferences among treatments allowing physicians and patients to select the most desirable one. GLINDA proposes a wide ontology of cross-guideline interactions (http://glinda-project.stanford.edu/guidelineinteractionontology.html). In GLINDA, one can express "timing interactions", representing scheduling inconsistencies. However, to the best of our knowledge, no work was published describing how "timing interactions" can be recognized and/or analysed using temporal reasoning techniques. We recently proposed an original approach, supporting user-driven and interactive interaction detection over different levels of abstractions [9], which is further described in Subsections 2.2.1 and 2.2.2. For a further comparison of some of these approaches (and others), see also [23].

None of the approaches coping with comorbidities in literature proposes temporal reasoning algorithms that support physicians in checking whether possible interactions between actions of different CIGs can actually occur in time. However, this is a major limitation of current approaches. Indeed, temporal issues are pervasive in the CIG context, and many previous approaches have demonstrated that their management is fundamental to provide substantial support to the execution of CIGs (see, e.g., the survey in [24]). In particular, in the Asbru [14] and in the GLARE [25] projects, rich representation formalisms have been proposed to cope with temporal constraints in the CIGs, and in GLARE we have proposed correct and complete temporal constraint propagation algorithms to reason with them and to merge them with the time of execution of actions on specific patients [25]. Combi et al. [26] proposed a conceptual model for clinical guidelines that considers

temporal constraints and their propagation, also in case of uncertainty and alternative paths. However, none of such approaches has considered temporal issues in connection to the treatment of comorbidities, which is the goal of the work in this paper.

2.2 GLARE and comorbidities: previous work

The approach described in this paper is based on GLARE. Starting from 2014, we began a project aiming at extending it to provide support to the management of comorbid patients. After a brief introduction of GLARE (our starting point), we sketch our previous work about the treatment of comorbidities, focusing on the ontology of interactions of GLARE (Section 2.2.1) and on the interaction detection approach (Section 2.2.2).

GLARE (Guideline Acquisition, Representation and Execution) [12] has been built starting since 1997 in a long-term cooperation between the Department of Computer Science of the University of Piemonte Orientale, Alessandria, Italy and the Azienda Ospedaliera San Giovanni Battista in Turin, one of the largest hospitals in Italy. GLARE supports the use of advanced artificial intelligence techniques and decision support techniques [13].

In GLARE, a CIG can be represented as a hierarchical graph, where nodes are the actions to be executed and arcs are the control relations linking them. GLARE distinguishes between *atomic* and *composite* actions (plans), where atomic actions represent simple steps in a CIG and plans represent actions that can be defined in terms of their components via the *has-part* relation.

GLARE adopts five types of atomic actions:

- *Work actions*, i.e., actions that describe an "external" procedure which must be executed (e.g., by a healthcare professional).

- *Pharmacological actions*, specifying a drug or a drug category to be administered to the patient and its dosage.

- *Decision actions*, used to model the selection among different alternatives.

- *Query actions*, i.e., requests of information (typically of patient's parameters).

- *Conclusions*, which explicitly identify the output of a decision action.

In this paper, we focus on composite actions, and on work and pharmacological atomic actions.

Actions in a CIG are connected through *control* relations. Control relations establish which actions can be executed next and in what order. In particular, the *sequence* relation explicitly establishes what the following action to be executed is; the *alternative* relation describes which alternative paths stem from a decision action and the *repetition* relation states that an action has to be repeated several times. The *constrained* relation is used in order to express more complex temporal relations between actions. In GLARE it is possible to express precise and imprecise dates, durations, delays and complex forms of repetitions [25]. For the sake of simplicity, in this paper we adopt an easier approach for repetitions: we assume that, when dealing with repeated actions, the exact number of repetitions is known and explicitly express the constraints between repetitions using the above language.

GLARE architecture is composed by two main modules: an acquisition module and an execution module. The acquisition module provides a graphical interface and different types of facilities (e.g., both syntactic and semantic checks about the consistency of the CIG) to support the acquisition of CIGs. Several CIGs in different domains, ranging from ischemic stroke to the emergency treatment of polytrauma, have been already acquired using GLARE. Acquisition is performed by a cooperation of a knowledge engineer (i.e., a GLARE expert) and a committee of physicians, experts in a given domain, and it may require days or weeks of work, depending on the complexity of the guideline. The execution module supports the execution of a specific guideline on a specific patient. Once again, a graphical interface is provided to support the user-physician. Knowledge engineers are not needed to support execution.

Starting from 2013, we are extending the GLARE approach to cope with comorbidities. The long-term goal of our overall approach is that of providing physicians with a domain-independent and guideline-independent set of tools and methodologies to facilitate the integration of two or more CIGs. Such a support may be adopted at different stages, i.e.:

1. To detect and analyze the interactions between two or more CIGs.
2. To solve the detected interactions.
3. To merge multiple CIGs in the treatment of a specific comorbid patient at hand.

The philosophy underlying our approach derives from our long-term interdisciplinary cooperation between computer scientists and physicians. We neither aim to provide any fully-automatic tool to synthesize the merging of two (or more) CIGs, nor we want to provide any tool intended to wholly take the place of physicians in any of her/his high-level activities. Indeed, our goal is that of providing a suite of tools and methodologies to support physicians, providing her/him, as much as possible, information and hints that may be helpful in her/his activity when facing multiple guidelines. This is also the underlying philosophy of the *mixed initiative* approach in artificial intelligence and human-computer interaction. In fact, Horvitz [27] defines mixed initiative as "*methods that explicitly support an efficient, natural interleaving of contributions by users and automated services aimed at converging on solutions to problems*".

Up to now, our work has mainly focused on the task 1. above, i.e., detection and analysis of interactions between the actions in two (or more) CIGs. In [9], we identified three different knowledge levels at which interactions might occur: (i) level of the goals of the CIG actions, (ii) level of the effects of the actions, and (iii) level of the drugs recommended by pharmacological actions. We have also pointed out that, in turn, all the levels may be structured at different degrees of detail. In [9], we have also proposed an ontological representation for the interactions at the different levels (see Section 2.2.1), and provided support for interactive physician-driven analysis of the interactions at the different levels (see Section 2.2.2).

2.2.1   An ontological model for interaction representation and recognition

Medical Decision Making is, in general, a process that involves different and heterogeneous knowledge sources, and the representation of such a knowledge is still an open problem in the Medical Informatics community. When facing the problem of the management of comorbidities, one of the most important sources to be considered is the basic medical knowledge regarding the interactions occurring between actions belonging to different CIGs. In a previous work [9], we focused on the modelling of such a knowledge and on the reasoning on it. In particular, we focused on interactions derived from the goals of the actions, from their effects and, in case of pharmacological actions, from the drugs they prescribe.

One of the basic elements of our model (see Figure 1) is the concept of *Variation*, which represents a change in the patient status. Each variation relates to exactly one *Attribute*, describing the patient's status, and to exactly one *Modality* of the variation. Attributes are modelled using SNOMED CT concepts [28], while modalities belong to the set {"*Increase*", "*Decrease*", "*Stability*"} and further refinements of its elements (e.g., "*Stop*" can be a refinement of "*Decrease*"). For instance, the variation "*Decrease Blood Pressure*" is modelled by the variation of the attribute "*Blood Pressure*" with modality "*Decreasing*". Variations are organized along a CAUSAL and PART-OF organization. Variations can cause other variations (e.g., the variation "*Decrease Blood Pressure*" can be caused by "*Decrease Blood Volume*", "*Inhibition of Angiotensin Converting Enzyme (ACE)*", "*Block of Calcium Channels*" and so on). Subcomponents of variations are modelled through PART-OF relations (e.g., "*Improve Lifestyle*" is decomposed into "*Improve Diet*", "*Increase Physical Activity*" and "*Stop Smoking*").

In our model, the effects of clinical actions (concepts *Work* and *Pharmacological Action*) are represented as variations (arc *hasEffect* in the figure). An action, regardless of the CIG in which it is recommended, is described by one or more effects. For instance, the action "*Administration of Aspirin*" is characterized by the effect "*Decrease Prostaglandin production*".

On the other hand, actions in a particular CIG (concept *CIGaction*, which is a subconcept of *Action*) are characterized by one or more goals, which are expected to be reached through the action execution. We call such goals *Intentions*, and we model intentions as a proper subset of *Variation*. Each CIG action is related in the ontology with one or more intentions (with the *aimsTo* arc). For instance, the action "*Administration of Aspirin*" is often used for the "*Antipyretic*" intention (i.e., to reduce the fever), however, in the prevention of heart attacks, it is administered with the "*Antiplatelet*" intention (i.e., to decrease platelet aggregation).

Actions that prescribe the administration of drugs (*Pharmacological Actions*) are characterized by the relation *substance* with the *drug* they recommend. Drugs are hierarchically organized – nodes are categories and leaves are specific drugs – and each level of the hierarchy is related (*hasEffect*) to its effects, which are defined as variations of the patient status. For the drug taxonomy, we use the ATC classification [29]; however, our approach is independent of the classification adopted.

We identify three types of interactions: interactions involving at least one intention (*IntentionInteraction*), interactions involving variations (*VariationInteraction*) and interactions involving drugs (*DrugInteraction*). In our model, all the interactions modify (*changes*) a variation, with a certain modality.

Variation interactions are described by the relation *hasElement*, with the two variations (or intentions) they involve and by a *Type*, whose basic values are *Concordance*, *Discordance* and *Independence*. However, further refinements are possible, such as *Opposite* for variations focusing on the same attribute, but discording in the modality. Intention interactions are a particular case of interaction between variations, in which at least one involved element is an intention.

Drug interactions are often caused by an interaction between two of the effects of the drugs they involve. In order to model such an information, the property *causedBy* relates a drug interaction to a *variation interaction*.

Following our model, two actions interact if two of their intentions, effects or drugs interact. However, interactions between intentions, variations and drugs are defined separately from CIGs. This allows a more structured representation of the knowledge and the *reusability* of such information. For instance, when a new CIG is introduced in the knowledge base, introducing the specifications of all the interactions between its actions and the CIGs already stored is not needed. At most, the ontological knowledge has to be extended with the description of the new actions (and variations and/or drugs) introduced by the new CIG and not already modelled in the ontology.
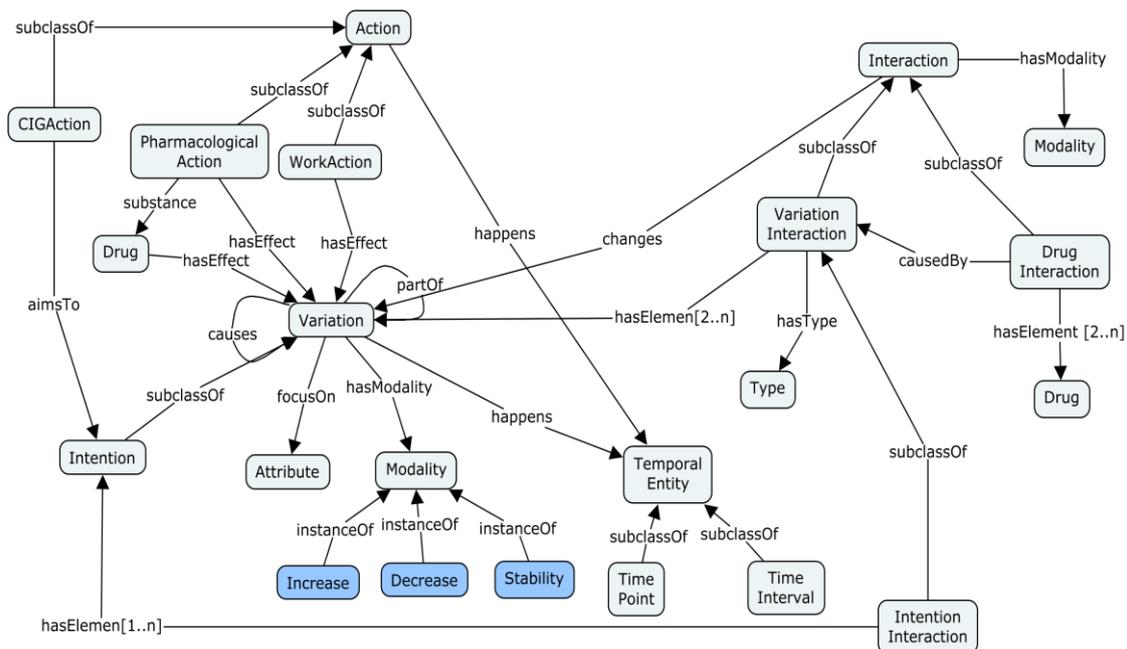

Figure 1. Schema of our ontological model. The left and right parts represent the non-temporal model of GLARE. Concepts *TemporalEntity*, *TimeInterval* and *TimePoint*, besides the arcs *happens*, are part of its temporal extension, which is detailed in Section 3.

2.2.2   Non-temporal detection of relevant interactions

Interactions between intentions, variations and drugs can be added to the previously described model in two ways: they can be (i) automatically detected by a reasoning tool or (ii) manually inserted in the ontology. Regarding point (i), knowledge regarding interactions between variations can be modelled through rules (see [9]) and a standard reasoner can be exploited in order to recognize interactions. In our implementation, we used OWL to represent our knowledge model (http://www.w3.org/TR/owl2-overview/) and we expressed rules to detect interactions using Semantic Web Rules (http://www.w3.org/Submission/SWRL/). With point (ii), we guarantee flexibility to the approach, allowing the experts to represent those interactions that are known in medical literature, but cannot be recognized by the reasoner (e.g., chemical interactions between drugs that do not depend on the effects they produce on the patient).

However, querying the ontology and simply giving as output all the interactions between all the actions belonging to the two CIGs could be not useful for physicians during the analysis. The output would contain too many elements, most of which would be irrelevant since interactions cannot effectively happen (e.g., in a specific patient some actions cannot be executed, thus all the interactions involving such actions are not relevant). For this reason, in [30] we detailed our mixed-initiative approach to the detection of *relevant* interactions between CIGs. Our approach allows physicians to navigate CIGs at different levels of abstraction, allowing them to focus on relevant parts of the CIGs and to compare only those actions considered important and potentially interacting.

Our interaction detection algorithm maintains the history of the focusing process, organized in a tree data structure (the navigation tree, see Figure 2). It supports both the addition of new CIG focuses and the rollback to upper focuses. Each node of the tree consists of three main components (for the sake of clarity, here we take into account only two interacting CIGs): two pairs, $\langle CIG_1, focus_1 \rangle$ and $\langle CIG_2, focus_2 \rangle$, determining the desired level of abstraction and the focused actions for the two CIGs, and an interaction component, in which, for each pair $\langle A_i, A_j \rangle$ of actions ($A_i \in focus_1$, $A_j \in focus_2$), the interactions between their intentions, effects or drugs they administer in the case of pharmacological actions are pointed out.

Furthermore, our approach, based on consensus and well-known medical ontologies and on a user-friendly interface, allows user physicians to navigate the ontology when an interaction is detected and to investigate how the system has recognized such an interaction.

Here we propose the non-temporal analysis for Example 1, introduced in Section 1, and Example 2, used throughout the rest of the paper. Notice that such examples are intentionally simplified in order to facilitate the understanding of the different phases of our approach.

Example 1 (Non-temporal analysis). Using the non-temporal detection of interactions, the physician navigates the two CIGs (represented, in a simplified version, in the node 1 of Figure 2), focusing first on the non-pharmacological treatment of the thrombosis CIG and on the antibiotic treatment part of the respiratory tract infection (see node 1.1). However, at this level, no interaction analysis is performed, so that in node 1.1 no interaction is reported. Then, s/he focuses on the Warfarin treatment (for the first CIG) and on the Erythromycin treatment (for the second CIG; see node 1.2). Then, s/he asks the system whether they interact. The system retrieves from the ontology the interaction between the two drugs and adds it to the node. However, such an interaction is only potential until the analysis does not consider time. Then, s/he focuses on the expansions of the Warfarin treatment and of the Erythromycin treatment (node 1.2.1). At such a level of detail, the CIG for thrombosis contains three Warfarin administrations, W1, W2 and W3. In the example, we suppose that the first two administrations have been already executed. On the other hand, the Erythromycin treatment contains two Erythromycin administrations, none of which has been executed yet. The physician may now want to know whether the last executed Warfarin administration (W2) can interact with the first Erythromycin administration that has to be executed next (E1). Our system detects such interaction from the non-temporal point of view. However, a temporal analysis is required to ascertain whether the interaction can occur in time, or it is only potential.
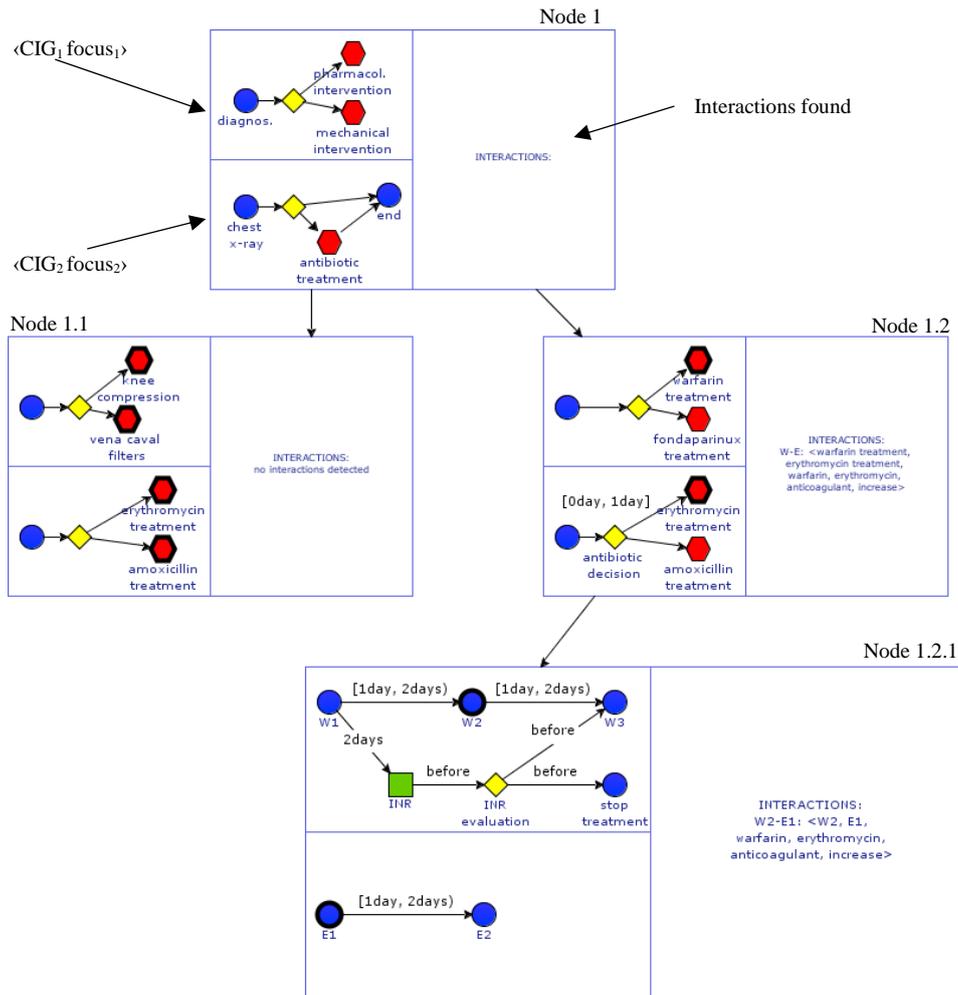
∎

Figure 2. Navigation Tree structure visualization for the non-temporal analysis of interactions.

Example 2 (Description and non-temporal analysis). In the second example, we consider a situation in which both Nalidixic Acid, an antibiotic used, e.g., for the treatment of urinary tract infection, and Calcium Carbonate, used for the treatment of the symptoms of gastroesophageal reflux, have to be orally administered to a patient. The result of their interaction decreases the antibiotic effects of the first. This is caused by the interaction between the urine alkalinization – effect of Calcium Carbonate – and the absorption of the Nalidixic Acid.

The process of non-temporal detection of this interaction may proceed as in Example 1. Further details can be found in [10]. In this situation, however, we suppose that the question asked by the physician is different. Since the interaction concerns the absorption of a drug, the physician may want to know if a temporal scenario for the actions execution may exist such that one can avoid the interaction. Also this question can be answered through temporal reasoning. ■

2.3 Limitations of current approaches

The treatment of comorbidities in the clinical practice necessarily involves the treatment of time, since physicians have not only to detect possible interactions between actions in different CIGs, but also to analyse whether such interactions can actually occur in time while treating the patient at hand. Indeed, actions whose effects do not overlap in time cannot lead to any *actual* interaction. The treatment of such a phenomenon requires:

> (i)   The representation of temporal information about actions, their effects and their interactions.
>
> (ii)   Temporal reasoning techniques propagating temporal information.

In summary, the literature overview above shows that, although several approaches in Medical Informatics cope with the representation of time, and some approaches also consider temporal reasoning, none of the current CIG approaches to comorbidities, including GLARE, considers such a fundamental aspect. In Table 1, for each main approach to comorbidities (see Section 2.1), we summarize its features using as comparison the support provided for the detection of interactions (we distinguish whether the approach takes in consideration temporal information) and the merge of the CIGs. For each approach, we indicate with an 'X' the fact that the approach provides support to a specific phase and with a '-' the fact that the approach does not provides support to the phase. The second to last row represents our previous work described in Section 2.2 and the last row of the table represents the approach presented in this paper. As can be clearly seen, most of the approaches in literature focus on the third phase (i.e., CIG merge), demanding to physicians the task of detecting interactions. Our approach and the one in Zamborlini et al. [16] are the only ones providing support to the detection of interactions. However, our approach is the only one that, exploiting temporal information, provides support to the temporal detection and analysis. Moreover, we have already started to extend GLARE to provide support also to the tasks of interaction resolutions and CIG merge (see [31,32]), aiming at becoming the first approach providing support to all the three phases of CIG integration identified in Section 2.2. Finally, it is worth stressing that our approach is the only one based on a mixed-initiative methodology, allowing an interleaving of contributions between human users and the system.

Table 1. Comparison with other approaches coping with comorbidities.

Approach	Detection		CIG Merge
	Non-temporal	Temporal	
Jafarpour and Abidi [20]	-	-	X
López-Vallverdú et al. [19]	-	-	X
Merhej et al. [21]	-	-	X
Sánchez-Garzón et al. [18]	-	-	X
Wilk et al. [17]	-	-	X
Zhang et al. [22]	-	-	X
Zamborlini et al. [16]	X	-	-
Piovesan et al. [9]	X	-	-
Anselma et al.	X	X	-

3 Temporal extension of the GLARE model

In this section, we extend the representation model of GLARE to include also the "*temporal*" knowledge. In Sections 4 and 5, we provide a temporal reasoning architecture that, exploiting such a knowledge, help the user physician in the temporal analysis of interactions.

In general, actions are characterized by the time of their occurrence. However, the time when an action occurs or is expected to occur is not sufficient to describe the entire phenomenon. Consider, for instance, the administration of a drug. The effects caused by the drug might not start immediately at the beginning of its administration and they do not usually end immediately with the end of the administration. The appearance of the effects and their duration can be usually estimated with some uncertainty starting from the time of administration. Such information can be usually found in the pharmacology literature and it is based on evidence.

In Section 3.1, we introduce our extension of the model to cope with temporal knowledge, first defining a general ontology for time and then expanding the GLARE nontemporal model for interactions. In Section 3.2, we describe the different sources of temporal knowledge we consider in our approach, i.e., CIGs, log and ontological knowledge.

3.1 Temporal ontology

Our temporal ontology grounds on several previous works about time in the Artificial Intelligence area, including [33,34], and, in particular, [35].

Our ontology meets two main requirements: it is expressive enough to represent the temporal aspects we focus on and (considering the temporal constraints) can be translated into STP (see Section 5.2).
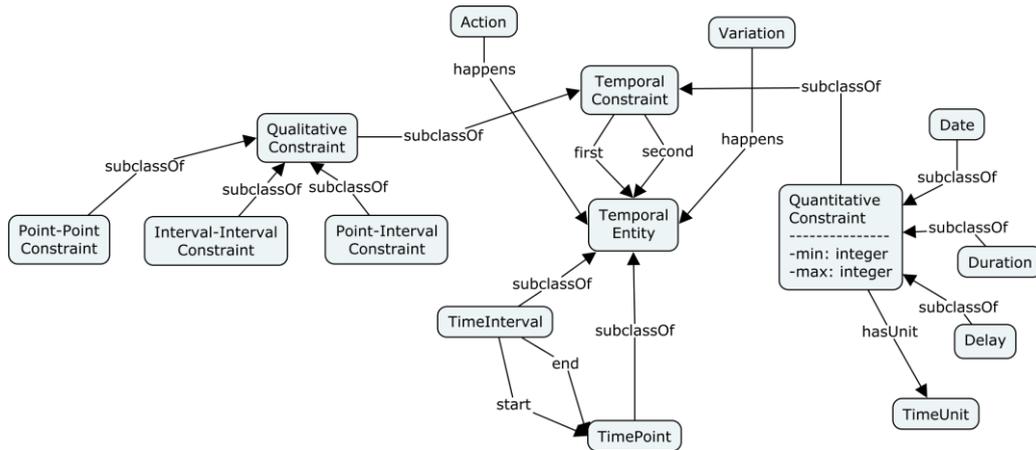
Figure 3. Schema of the ontology of time and time constraints.

In our temporal model (see Figure 3), we admit two basic *Temporal Entities*: *Time Intervals* and *Time Points*. A time interval is itself described by two time points, which represent its starting and ending times.

We use temporal constraints between pairs of temporal entities (arcs *first* and *second* in Figure 3) to represent the temporal relations between them. In particular, constraints can be *qualitative* or *quantitative*.

Qualitative constraints represent the relative position of two temporal entities. We use a fragment of the algebra in [33] to represent qualitative relations between time points and time intervals. Vilain distinguishes between point-point constraints (●Before●, ●After●, ●Equals●), point-interval constraints (●Before, ●Begins, ●During, ●Ends, …) and interval-interval constraints (Before, After, Ends …). The latter are usually known as Allen's basic temporal relations [34]. Vilain considers also arbitrary disjunctions of such constraints. On the other hand, in our approach, we restrict our attention to the subset of Vilain's constraints that can be expressed in STP, i.e., on the *continuous pointizable* constraints [36].

Quantitative constraints consider *metric* time. In our approach, they are characterized by a *minimum* and a *maximum* value bounding the span of time between two time points and a *time unit* for the bounds. It is worth stressing that such a representation supports ***temporal indeterminacy***, since the maximum and minimum values can be different. We identify three basic types of quantitative constraints: *Date*, *Delay* and *Duration*.

Date: a date models an "absolute" metric time. Indeed, a date can be represented by a constraint between a time point and a *reference time*, which is a fixed point on the timeline unique for all the dates.

Duration: a duration characterizes a time interval representing the distance between its ending and starting points.

Delay: a delay represents the distance between two general time points.

In our extended ontology (see Figure 1), actions, variations and intentions are related to a temporal entity through the arc *happens*. Such arc assumes different meanings, depending on the type of the concept it starts from:

- For CIGactions it represents the time when the action is executed. An action happening in a time interval is a *durative* action, while an action happening in a time point is a *punctual* one.
- For effects (or other variations that are not intentions) it represents the time when the effect should happen.
- For intentions it represents the time when the physician expects the intention will be accomplished.

3.2 Sources of temporal information

In our approach, we consider different sources of temporal information (CIGs, log and ontological knowledge). CIGs represent the general "plans" that should be executed, providing information regarding actions and their order of execution; logs contain the information regarding the parts of

the CIGs already executed on a specific patient; the ontological knowledge contains information concerning the interactions between actions. In this section, we describe how we cope with such sources, with specific attention to the temporal aspects. Notably, all the temporal constraints considered below are based on the temporal ontology in Section 3.1, and – as discussed in Section 5.2 – can be mapped onto STP.

CIG temporal knowledge. The type of temporal constraints contained in CIGs strictly depends on the specific representation formalism being adopted. In GLARE and in many other formalisms, to maintain knowledge regarding the order of execution of the actions, CIGs directly contain qualitative temporal constraints between the actions as well as quantitative temporal constraints about their duration and about delays between them [25]. GLARE acquisition module supports the acquisition of such temporal constraints by a team of knowledge engineers and physicians, through a graphical interface.

Example 1 (CIG temporal knowledge). In Figure 4, for instance, we report the temporal information contained in the expansion of the composite action "Warfarin treatment" of the CIG for venous thrombosis. Warfarin is usually administered once a day. Thus, the CIG for thrombosis contains temporal constraints between consecutive warfarin administrations (W1, W2 and W3) binding the delay between them to be greater than or equal to one day and to be less than two days. In Example 1, we work at the granularity of days. ∎
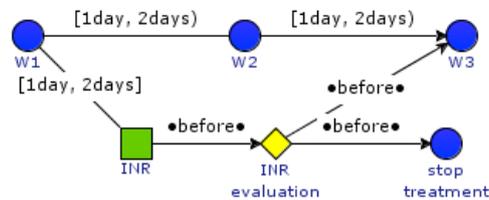

Figure 4. Expansion of the composite action "Warfarin treatment" of the CIG for thrombosis prevention. For the sake of simplicity, we report only three repetitions for the action "Warfarin administration", represented by the actions W1, W2 and W3.

Log knowledge. Given a patient, a log contains information about the CIG actions executed on her/him. Therefore, a log identifies the path executed until now in the CIG. Depending on the context, logs can contain more or less detailed temporal information about the time of execution of such actions. For the sake of generality, in our approach we distinguish among three different types of logs.

1. *Temporally exact log*. In this context, the times when the actions of the CIGs have been executed on the patient are known and have been recorded with the precision allowed by the granularity chosen for the log (e.g., hours or minutes). For instance, in monitoring contexts such detailed temporal information may be available.
2. *Temporal imprecise log*. In many medical contexts, it is not realistic to assume to know the exact execution time of each action. In the context of temporal imprecise log, we allow imprecision in time measurements. In fact, the log could contain not the exact start/end time of the clinical actions but, e.g., a time interval when the actions started and ended and/or information about the relative temporal placement of the actions (e.g., qualitative constraints such as "before"), and/or information about delays.
3. *Non-temporal log*. In this context, the log lists the executed actions without any additional temporal information.

Example 1 (LOG knowledge). In Table 2, we report some examples of log entries for the Warfarin administrations of Figure 4 in case of temporally exact, temporally imprecise or non-temporal log. Notice that, in our example, actions W1 and W2 have already been executed, while W3 has not yet been executed. Henceforth, we consider as a running example the case of temporally imprecise log.

■

Temporally exact log	Temporal imprecise log	Non-temporal log
▪ Warfarin treatment started on 01/04/2015 ▪ W1 executed on 01/04/2015 ▪ W2 executed on 02/04/2015	▪ Warfarin treatment started on 01/04/2015 ▪ W1 executed on 01/04/2015 ▪ W2 executed ●after● W1	▪ Warfarin treatment started ▪ W1 executed ▪ W2 executed

Table 2. Examples of the three different types of log supported by our approach.

Ontological temporal knowledge. The ontological knowledge contains temporal information about the interacting elements. In particular, as previously explained, the ontological knowledge models the delay between action execution and the appearance of the respective variations (intentions, effects and effects of the drugs) and the durations of such variations. In some cases, variations can be persistent (i.e., they do not end, unless some other actions or external factors change the status of the patient). In such a case, their duration maximum value is the special value *maxint*. Such a knowledge is modeled by a joint team including both expert physicians and knowledge engineers, on the basis of the medical literature, and using the Protégé tool [37] to support the process.

Example 1 (Ontological temporal knowledge). In Figure 5, we illustrate how we model temporal knowledge about the "*Anticoagulant*" effect of the action "*Warfarin Administration*". An action of type "*Warfarin Administration*" is "punctual", i.e., it happens in a time point W_p, and it is characterized by the "*Anticoagulant*" effect. Such an effect happens in a time interval AN_I, starting at the time point AN_S and ending at AN_E, which is expected to begin in at maximum one day after the administration (modelled through the constraint "*WAAN_{delay}*") and it usually lasts from one to four days (modelled through the constraint "*ANTI_{duration}*"). ■
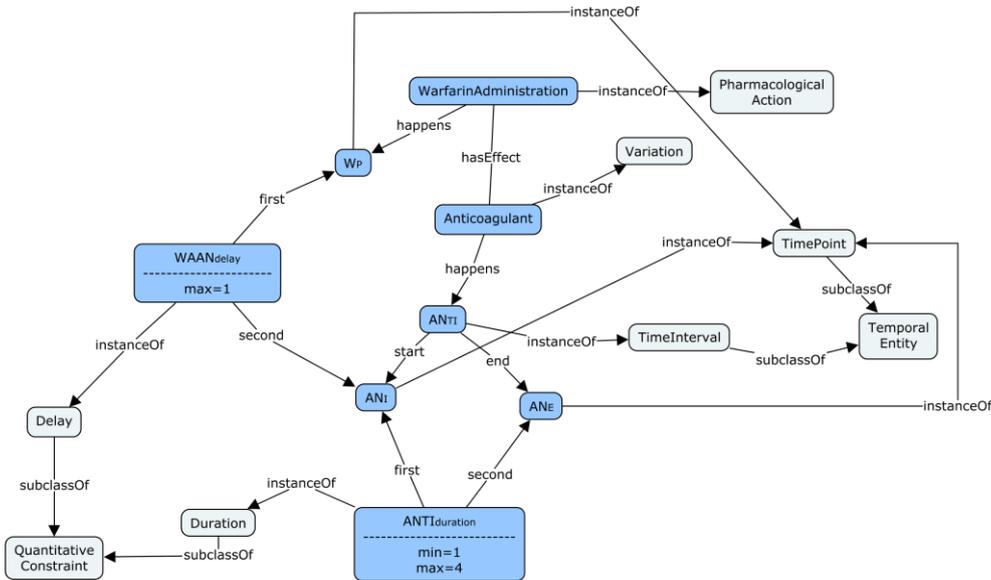

Figure 5. Representation of temporal constraints for the "Anticoagulant" effect of the action "Warfarin Administration", following the previous described model.

4 Temporal reasoning for comorbidities

To support users, the above knowledge model must be complemented by (i) reasoning algorithms working on it and (ii) user-friendly facilities for the analysis of interactions. To guarantee the maximum flexibility, our approach provides user-physicians with different options for the analysis. We provide high-level and user-friendly facilities to users, and an interactive graphical tool, thus hiding to users the internal complexity of the temporal reasoner, which operates in a fully automatic way.

In this section, we just propose a high-level view of our temporal approach, describing how the temporal extension presented in this paper is integrated with the non-temporal detection analysis

(Section 2.2), and the set of facilities we provide to physicians to deal with temporal interactions in different application scenarios (Section 4.2). The technical treatment of how the facilities can be provided is demanded to the next section (Section 5).

4.1 Overall approach to interaction analysis

Before delving into the discussion on the temporal reasoning algorithms, it is important to describe how the user and the system interact to detect and analyse the CIG interactions, following the mixed-initiative paradigm.
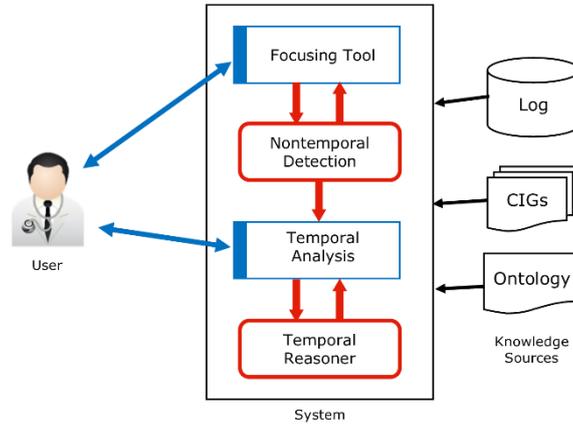

Figure 6. Schema representing the modules of our system and their interaction with the physician. Blue rectangles (with thicker left margins) represent "interactive" modules, while red ones (with rounded edges) represent fully automatic ones.

In Figure 6, we schematize the interaction of physicians with our system during the analysis process. Our system is constituted by four main modules. Two of them (blue rectangles with thicker left margins) require the interaction with the user, while the other two (red rectangles with rounded edges) operate in a fully automatic way.

The two upper modules ("Focusing Tool" and "Nontemporal Detection") concern the non-temporal analysis phase, described in Section 2.2.2. Basically, the high-level module "Focusing Tool" shows the navigation tree to the user and provides several facilities to explore and expand it. On the other hand, the "Nontemporal Detection" module navigates the ontology to retrieve all the interactions involving the intentions/effects of the focused actions. The output of the non-temporal analysis phase is a new focus in the navigation tree (i.e., a set of actions in the CIG at a level of abstraction chosen by the user-physician) enriched with the list of the detected interactions (consider, e.g., the running examples in Section 2.2.2). Such interactions are, however, only potential, since they have been detected disregarding the temporal dimension. Thus, once a list of potential interactions has been obtained, the physician can decide to temporally analyze one (or more) of them in order to know whether such interactions can actually occur, on the basis of the temporal constraints imposed by the different sources of temporal information (i.e., CIGs, logs and ontological knowledge). Again, this phase is characterized by an interleaving of contributions by the user and by the system. In particular, interacting with the "Temporal Analysis" module, the physician can choose among different types of analysis (see the facilities presented in Section 4.2). For instance, s/he may ask the system whether the interaction can occur in time, or a temporal scenario for the execution of future actions to enforce (or avoid) an interaction, possibly considering additional temporal hypotheses. Given a user query, the system operates in a fully automatic way, extracting the temporal constraints from the input sources, translating them into STP and finally propagating the constraints. As discussed in Section 5.3, the output of constraint propagation is a *minimal network*. However, as reported in , we hide to user physicians the "Temporal Reasoner" module and the low-level structure of the minimal network. Indeed, the minimal network is returned to the "Temporal Analysis" module, which provides the user with a graphical interpretation of the results, allowing her/him to refine and/or explore the output of the reasoner.

Obviously, both the non-temporal and the temporal analyses can be performed several times at different levels of abstraction and considering different facilities, until the user physician is satisfied by the obtained result.

4.2 Temporal facilities

In order to cope with the interactions between CIGs, we devised several temporal facilities to be provided to user-physicians. Such facilities allow a user to temporally analyze interactions, possibly considering some hypotheses, e.g., on the time of execution of some actions in the CIGs. It is worth noting that, even if in most cases medical interactions are considered "adverse" elements to be avoided, in some cases they can be desired by the physician. Consider, for instance, the case in which a patient treated for hypertension suffers also from edema. The physician may desire the interaction between the diuretics prescribed for edema and the antihypertensive drugs since the interaction between the two drug categories enhances the effect of the latter. For this reason, we support also the possibility to "obtain" an interaction.

The temporal facilities are:

1. *Interaction.* It checks whether two actions in two different CIGs may interact, necessarily interact or certainly do not interact.
2. *Interaction (what-if).* It allows a user to assume a hypothetical execution time for some future actions and to ascertain the effects of such an assumption on the interactions between two CIGs.
3. *Time of future actions to avoid (or to have) an interaction.* It shows when some future actions have to be executed to avoid or to achieve some interactions.
4. *Time of future actions to avoid (or to have) an interaction (what-if).* It allows a user to make some assumptions concerning the time of some future actions and to determine their effect on the times of the future actions in order to avoid or to have some interactions.

Definition. We call *focused future actions* the actions that, in a path of a CIG, occur between the last-executed action (or the first action of the CIG in case of no log scenario) and the actions involved in the analyzed interaction. Notice that, while the log determines a single path in the CIG considering actions executed in the past, future actions can contain alternative paths. Each path is analyzed by our approach independently of the others, unless some paths are explicitly excluded through a graphical interface by the user physician. ∎

Notice that Facilities 1 and 2 might provide not-crisp answers, in the sense that an interaction between two actions can be temporally necessary, temporally possible or temporally impossible. On the other hand, Facilities 3 and 4 should return a temporal scenario for the execution of future actions.

Example 1 (Facilities). As previously discussed, in Example 1 the physician is interested in the interaction between the Warfarin and the Erythromycin treatments. Specifically, s/he may want to know whether the last-executed Warfarin administration W2 can interact with the first (not performed yet) administration of Erythromycin E1. Notably, such an interaction is potential, in the sense that it has been detected by the non-temporal analysis – the output of such an analysis was the detection of "W2-E1"; see Section 2.2.2. Now, s/he is looking whether such an interaction can occur in time. Such a task can be achieved through Facility 1 (*Interaction*).

```
(Q1) CHECK TEMPORAL INTERACTION ("W2-E1")
```

Notably, in this context, also Facility 2 (*Interaction (what-if)*) may be useful. For instance, the physician can express the hypothesis that the first action of the Erythromycin treatment (E1) happens the same day of W2:

```
(Q2) CHECK TEMPORAL INTERACTION ("W2-E1")
        IF (HAPPENS(E1) ●EQUALS● HAPPENS(W2))
```

∎

Example 2 (Facilities). In the case of Example 2, the physician is interested in the time of future actions to avoid the interaction between Nalidixic Acid (NA) and Calcium Carbonate (CC), called "NA-CC". Furthermore, the physician hypothesizes that the administration of Calcium Carbonate happens between 12pm and 2pm. Such a task can be achieved through Facility 4.

```
(Q3) AVOID TEMPORAL INTERACTION ("NA-CC")
        IF (HAPPENS(CC) ●DURING [01/04/2015 AT 12, 01/04/2015 AT 14])
```

∎

The analysis of interactions can be performed in different settings, as mentioned in the introduction. We singled out four different scenarios in which we support the temporal analysis of interactions, along two different dimensions:

- *Patient-independent* vs *patient-dependent* analysis of interactions;
- Availability and nature of temporal information in the *logs*.

Patient-independent analysis is the analysis of the "abstract" interactions between CIGs, not taking into account any specific execution of CIGs on specific patients. In such a case, no log is available since no specific execution is considered. On the other hand, patient-dependent analysis focuses on the interaction between CIG executions, considering a specific patient. In particular, depending on the availability and precision of temporal information in the log, we consider, as discussed in Section 3.2, (i) non-temporal, (ii) temporally exact and (iii) temporally imprecise logs.

As a consequence, four different scenarios are supported by our approach:

(1) Patient-independent analysis (no log).
(2) Patient-dependent analysis with non-temporal log.
(3) Patient-dependent analysis with temporally imprecise log.
(4) Patient-dependent analysis with temporally exact log.

These scenarios have different impacts on the facilities available to the user-physicians. In the following, we introduce the temporal reasoning algorithms we provide to cope with each one of the different possibilities, i.e., for each pair <facility, scenario>. Indeed, instead of proposing a heterogeneous set of solutions and providing a different approach for each one of the possible pairs, we adopt a modular and compact approach. We:

(i)   Identify and develop the different basic reasoning capabilities needed to solve each task.
(ii)   Show, for each case (i.e., for each pair <facility, scenario>) the combination of basic capabilities needed to manage it.

Such results are schematized in Table 5 in Section 5.5, where we report, for each type of scenario and for each facility, the reasoning and interpretation tools used to manage them.

5 The temporal reasoning engine

In this Section, we show in technical detail how the different facilities discussed in Section 4 can be provided. We first describe the overall architecture of our temporal approach (Section 5.1). Then, we describe the modules constituting such an architecture. Section 5.2 shows how constraints are collected from the different sources and are translated into STP. In Section 5.3, we show how STP reasoning is extended to cope with the temporal phenomena involved by interaction analysis and in Section 5.4 we explain how the resulting minimal network is interpreted, depending on the selected facility, to return user-friendly results to user physicians. Finally, in Section 5.5 we describe how the different facilities are supported on the basis of our extended temporal reasoner and of our interpretation mechanism.

5.1 Temporal reasoner architecture

In analogy with the non-temporal architecture, we define two main modules to manage the temporal extension of our system (see Figure 7). The first one, the "Temporal Analysis" module, provides a high-level interface to communicate with users, hiding them the details of the underlying second module, i.e., the "Temporal Reasoner" module. Both modules are further structured into sub-modules. In this section, we briefly present the organization of such submodules, while, in the
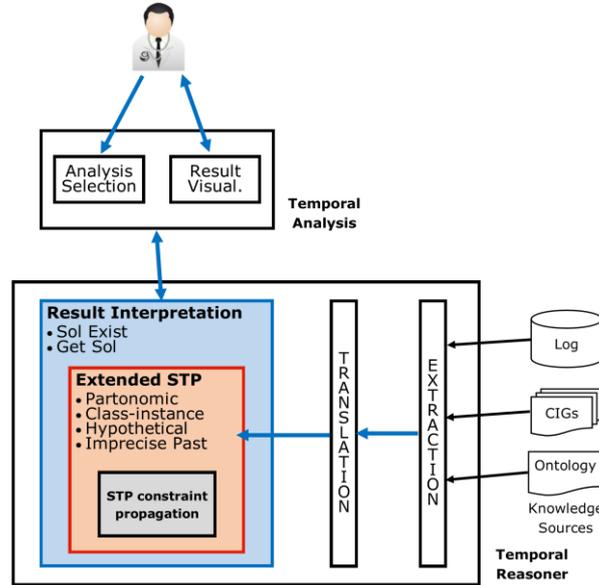

Figure 7. Architecture of the temporal modules. This architecture represents the expansion of the "Temporal Analysis" and "Temporal Reasoner" modules of .

following sections, we describe in detail each of them.

Two submodules manage the interaction between users and the system. Given a selected interaction, the "Analysis Selection" module allows physicians to temporally analyze it, choosing among different analysis options (see the facilities in Section 4.2). On the other hand, the "Result Visualization" module operates when an output of the temporal reasoner has to be presented to users (see Section 6). Notably, the interaction between such a module and physicians is bidirectional: indeed, in the following, we will show that physicians can contribute to refine the results returned by the temporal reasoner.

The Temporal Reasoner module is subdivided into five submodules. Given a selected facility, "Extraction" and "Translation" (see Section 5.2) collect the temporal constraints from the log, the CIGs and the ontology and translate them into STP the temporal constraints. The remaining three modules are organized in a "multi-layered structure". The innermost module is the STP constraint propagation framework. The "Extended reasoning" module extends it to provide more complex reasoning tasks, required to support the user facilities (Section 5.3). Then, the outer layer ("Result Interpretation" module; explained in Section 5.4) interprets the resulting minimal network. The selection of the extended reasoning algorithms and the interpretation of the results depend on the selected facility and on the nature of the execution log. In Section 5.5, we describe how such a selection is performed.

5.2   Constraint extraction and translation

Temporal information derives from different sources: the log of execution of the actions (when it exists), the CIGs and the ontology. The preliminary steps to perform temporal reasoning are the collection of such pieces of information from the different sources and their translation into STP (omitted –for the sake of brevity– from Table 5). These two phases are common for each facility we provide.

The goal of the extraction module is to extract all and only those temporal entities (i.e., intervals of occurrence of actions, intentions and variations – see the ontology in Figure 1) and constraints that are strictly required. To this purpose, we devise an algorithm that, starting from the temporal entities

involved in an interaction, navigates backward and across the knowledge sources, retrieving all and only the temporal constraints that can influence them.

The input of our extraction process are the three knowledge sources – log, ontological knowledge and CIGs – and an interaction found through the non-temporal procedure. In our algorithm (see the algorithm "extraction" in the following), we call such interaction *inter*, and ia_1, ia_2, … ia_n the CIGactions it involves (*interacting actions*), where ia_j denotes the interacting action belonging to CIG$_j$. The output of the extraction algorithm is a collection of constraints (*constr*), each of them tagged with a label specifying the source it comes from, i.e., ONTO for "ontological knowledge", LOG and CIG.

The process of extraction is divided into two phases. The first phase extracts *from the ontology* all the temporal constraints of the ontological elements (intentions, effects or drugs) involved in the interaction *inter*. As discussed in Section 3, such elements are related through temporal constraints to the times of execution of the interacting actions ia_1, ia_2, ..., ia_n. The second phase retrieves *from the log and the CIGs* all the temporal constraints that bind in some way the execution of the actions *ia*. To do this, we define a set of actions *fa* (*focused actions*), initially containing only the actions ia_1, ia_2, ..., ia_n. The algorithm iteratively considers each action *act* of *fa* and extracts from the log and the CIGs the temporal constraints concerning the execution of such an action. If some of the retrieved constraints involve an action that is not contained in *fa*, such an action is added to the set. The algorithm ends when all the actions in *fa* have been considered. When the algorithm ends, *fa* contains all the actions that influence the time of execution of the actions ia_1, ia_2, ..., ia_n.

---
extraction algorithm

```
input: CIGs, log, ontology, Interaction inter, set<CIGaction> {ia₁, ia₂, …
 iaₙ};
output: set<Constraint>;
begin
  set<Constraint> constr ← {};
  if DrugInteraction(inter)
  begin
    inter ← causes(inter);
  end
  constr ← constr ∪ ontoextract(ia₁, ia₂, … iaₙ, inter, ontology);
  set<CIGaction> fa ← {ia₁, ia₂, … iaₙ};
  foreach act in fa do
  begin
    <newconstr, newactions> ← logCIGextract(act, log, CIGs);
    constr ← constr ∪ newconstr;
    fa ← fa ∪ newactions;
  end
  return constr;
end
```

In the following, we explain how we extract temporal constraints from the ontology, the log and the CIGs.

Extraction from the ontology (*ontoextract(ia₁, ia₂, … iaₙ, inter, ontology)*). For each ontological element (intention/effect) involved in the considered interaction *inter*, we extract from the ontology each temporal constraint characterizing it (e.g., the duration of its temporal entity) and its relationship with the corresponding action ia_j (e.g., the delay between the action execution and the beginning of the time entity of the intentions/effect). In the case of drug interactions, we consider the *variation interaction* reachable through the *causes* arc (*inter ← causes(inter)* – see the ontology in Figure 1). All the constraints extracted during this phase are tagged with the label ONTO.

Extraction from log and CIGs (*logCIGextract(act, log, CIGs)*). In the scenarios where a log exists, we extract from it the temporal constraints regarding the starting or ending points of *act*. Such constraints are tagged with the LOG label. On the other hand, from CIGs we extract the temporal constraints imposed by *control* arcs between the considered action *act* and other actions belonging to the same CIG tagging them with the label CIG. If, during the collection of temporal constraints regarding *act*, an action is found, it is added to a temporary set (*newactions*). Such an operation guarantees that, if an action influences the time of execution of *act*, its temporal constraints have been already extracted or will be extracted before the end of the process. Both the set of extracted

constraints (*newconstr*) and of actions (*newactions*) are returned. In the case of temporally exact log, a refinement of the extraction process is possible. Indeed, if the log contains two exact dates regarding the starting and ending points of the execution of *act*, *act* is fixed in time, i.e., we have exact and complete information about its execution. For actions fixed in time, the extraction of the temporal constraints in the control arcs from the CIG is not needed, since they do not add any additional information.

Once the relevant temporal information has been extracted, we translate it into STP. In Table 3, we describe the different types of constraints that are expressible in our approach and how they are translated into STP. As discussed in Section 3, such constraints can be quantitative (e.g., durations) or quantitative (e.g., the "after" relation) and they can relate points or intervals. For each constraint, we report an example in natural language, its translation to STP and the example translated to STP. In the table, we assume a reference time RT=1/1/2015 00:00 that anchors the events to a calendar and we use minutes as the basic granularity.

It is worth noticing that in our approach only the qualitative constraints that can be mapped to STP can be used [25]. The translation to STP of each constraint requires a constant time.

Notation. Henceforth, given an action Act, if Act is punctual, Act_P represents the time point in which Act is executed. If Act is durative, Act_S represents its starting point and Act_E is its ending point. ∎

Example 1 (Extraction and Translation). Considering a granularity of days and reference time RT = 01/04/2015 (corresponding to the time of execution of the first Warfarin administration), the STP constraints derived from the Warfarin administrations in the CIG for venous thrombosis of Example 1 are

$[0 \leq AN2_S - W2_P \leq 1]_{ONTO}$	(C1)	$[1 \leq AN2_E - AN2_S \leq 4]_{ONTO}$	(C2)
$[0 \leq W1_P - RT \leq 0]_{LOG}$	(C3)	$[0 < W2_P - W1_P \leq +\infty]_{LOG}$	(C4)
$[1 \leq W2_P - W1_P < 2]_{CIG}$	(C5)		

where Wn_P is the time of execution of the nth warfarin administration, ANn_S and ANn_E are the endpoints of the anticoagulant effect of the nth warfarin administration, RT is the reference time, En_P is the time of execution of the nth erythromycin administration, and RMn_S and RMn_E are the endpoints of the reducing metabolism effect of the nth erythromycin administration.

Constraints C1 and C2 are extracted from the ontological knowledge (see Figure 5) for the interacting action W2, while constraints C3, C4 and C5 are extracted by the procedure *logCIGextract* (see Figure 4 and the second column of Table 2).

Similarly, constraints regarding the CIG for respiratory tract infection are collected and translated, obtaining the following:

$[0 \leq RM1_S - E1_P \leq 1]_{ONTO}$	(C6)	$[2 \leq RM1_E - RM1_S \leq 6]_{ONTO}$	(C7)

∎

Example 2 (Extraction and Translation). Considering a granularity of hours and a reference time RT = 0, the STP extracted from Example 2 is the following:

$[4 \leq UA_E - UA_S \leq 5]_{ONTO}$	(C8)	$[0 \leq UA_S - CC_P \leq 1]_{ONTO}$	(C9)
$[1 \leq NAA_S - NA_P < 1]_{ONTO}$	(C10)	$[4 \leq NAA_E - NAA_S \leq 4]_{ONTO}$	(C11)
$[0 \leq NA_P - RT < 24]_{CIG}$	(C12)		

where CC_P and NA_P are the times of execution of the Calcium Carbonate and Nalidixic Acid administrations, UA_S and UA_E are the endpoints of the urine alkalinization effect of the Calcium Carbonate, and NAA_S and NAA_E are the endpoints of the Nalidixic Acid absorption.

Constraint	Example	Translation to STP	Example translated to STP
Quantitative constraints			
precise absolute times (P)	P happens on 1/1/2015 at 10:00	distance from RT	$600 \leq P - RT \leq 600$
precise absolute times (I)	I starts on 1/1/2015 at 10:00 and ends on 1/1/2015 at 12:00	distance between RT and the starting point of I and between RT and the ending point of I	$600 \leq I_S - RT \leq 600$ $\wedge$ $720 \leq I_E - RT \leq 720$
imprecise absolute times (P)	P happens on 1/1/2015 in a time between 10:00 and 11:00	distance from RT	$600 \leq P - RT \leq 660$
imprecise absolute times (I)	I starts on 1/1/2015 between 10:00 and 11:00 and ends on 1/1/2015 between 12:00 and 13:00	distance between RT and the starting point of I and between RT and the ending point of I	$600 \leq I_S - RT \leq 660$ $\wedge$ $720 \leq I_E - RT \leq 780$
precise durations (I)	I lasts one hour	distance between endpoints of I	$60 \leq I_E - I_S \leq 60$
imprecise durations (I)	I lasts between one and two hours	distance between endpoints of I	$60 \leq I_E - I_S \leq 120$
precise delays (P)	P2 happens 30 minutes after P1	distance between P1 and P2	$30 \leq P2 - P1 \leq 30$
imprecise delays (P)	P2 happens 20-50 minutes after P1	distance between P1 and P2	$20 \leq P2 - P1 \leq 50$
precise delays (I)	I2 starts 30 minutes after I1	distance between the ending point of I1 and the starting point of I2	$30 \leq I2_S - I1_E \leq 30$
imprecise delays (I)	I2 starts 20-50 minutes after I1	distance between the ending point of I1 and the starting point of I2	$20 \leq I2_S - I1_E \leq 50$
Qualitative constraints			
point-point constraints	P2 is after P1	distance between P1 and P2	$0 \leq P2 - P1 < +\infty$
interval-interval constraints	I2 is during I1	distance between endpoints of I1 and I2	$0 < I2_S - I1_S < +\infty \wedge$ $-\infty < I2_E - I1_E < 0$
point-interval constraints	P starts I	distance between P and starting point of I	$0 \leq I_S - P \leq 0$

Table 3. Summary of the temporal constraints we support and their translation to STP. Please note that P stands for point and I stands for interval, and that, for an interval I, I_S denotes its starting point and I_E its ending point. In the table, we adopt minutes as the basic temporal granularity and RT=1/1/2015 00:00 as reference time.

5.3 Extended STP reasoning

Different forms of reasoning are required to provide the facilities introduced in Section 4.2. While STP is a fundamental block, it is not sufficient to provide all the temporal facilities. In particular, the treatment of temporal interactions between CIG actions demands the addition of four types of temporal reasoning that are not directly supported by STP:

(i)   Partonomic (temporal) reasoning, to support the treatment of constraints deriving from the fact that CIG actions may be composed by sub-actions.

(ii)   Class-instance (temporal) reasoning, to support the fact that both general constraints between "classes" of actions (i.e., the constraints among actions in the CIGs) and

specific constraints about "instances" of actions in the log have to be managed together.

(iii) Hypothetical (temporal) reasoning, to infer the consequences of temporal hypothesis.

(iv) Non-restriction reasoning, to check that temporal hypothesis do not improperly restrict the time of already performed actions.

In our approach, we provided and extended STP reasoning module, containing sub-modules copying with such aspects. In the following, we discuss each one of the extensions we provided. However, before going into the technical details, it is worth stressing that, in all the different cases, we manage the reasoning problems in such a way that they can be solved with the application (maybe multiple application, on different set of constraints) of Floyd-Warshall's algorithm to STP constraints. In such a way, we can grant that our temporal reasoning approach is both *correct* and *complete*.

5.3.1 Partonomic temporal reasoning

Partonomic relations in the CIGs (i.e., the fact that composite actions can be defined in terms of their components) induce temporal constraints that have to be represented in the STPs. Indeed, this phenomenon can be easily managed on top of STP. In particular, when an action B is part of an action A we have the STP constraints $0 \leq B_S - A_S \leq +\infty$ and $-\infty \leq B_E - A_E \leq 0$. For instance, in Example 1, the action E1 is part of the high-level action Erythromycin treatment. Then, our system extracts also the partonomic constraints $[0 \leq E1_P - ET_S \leq +\infty]_{CIG}$ and $[-\infty \leq E1_P - ET_E \leq 0]_{CIG}$ where ET_S and ET_E are the ending points of the action Erythromycin treatment.

5.3.2 Class-instance reasoning

The temporal constraints in the CIGs can be considered constraints on *classes* of events, which are instantiated each time a CIG is executed on a specific patient. Thus, the problem of checking whether a specific execution of a CIG satisfies the temporal constraints of a CIG corresponds to checking whether the temporal constraints of the instances satisfy the temporal constraints of the classes. This problem has been dealt with in [38]. In that work, the authors have considered the problem of "inheriting" the temporal constraints from classes of events to instances of events. They have also singled out two issues that, in general, make the inheritance a difficult problem: *correlation* and *observability*. Given a temporal constraint between two classes of actions, correlation concerns the issue of identifying the corresponding pair of instances of such actions that have to inherit the temporal constraint. Observability concerns the issue of knowing whether and/or to what extent one can assume that the executed actions are indeed observed and recorded. In [25] the authors have extended such an approach in order to support also periodic events and they considered the domain of CIGs. As stated in [25], in the CIG domain, which is also the context we consider in this paper, it is possible to assume that correlation is exactly known and that there is full observability of the instances. Therefore, as discussed in [25], it is possible to cope with the class-instance temporal reasoning by merging in a single STP both the constraints between classes and the constraints between instances, and then performing STP temporal reasoning. For each pair of time points, STP can take into account just one minimum and one maximum distance; in case both the CIG and the LOG provide such pieces of information, they have to be "merged" by considering the maximum of the minimum distances and the minimum of the maximum distances. For instance, given two time points A and B, and two constraints $[c_1 \leq B - A \leq d_1]_{CIG}$ and $[c_2 \leq B - A \leq d_2]_{LOG}$, the "merged" constraint to be inserted in the STP is $[\max(c_1, c_2) \leq B - A \leq \min(d_1, d_2)]_{CIG,LOG}$. In our example, constraints C4 and C5 are both referred to the difference $W2_p - W1_p$. Then, the resulting constraint is $[1 \leq W2_P - W1_P < 2]_{CIG,LOG}$.

5.3.3 Hypothetical reasoning

"Hypothetical reasoning" is used to introduce new constraints into a given set of constraints and to investigate the effects of such an addition. Hypothetical reasoning is used in two different cases in our approach, i.e., to manage both *explicit* and *implicit* hypothetical constraints. *Explicit* hypothetical constraints are tagged with the label "HYP" and are introduced by user-physicians when adopting the facilities 2 and 4 (see Table 5), in which the users can investigate the temporal effects of assuming hypotheses on the time of execution of *future* actions. In query Q3, for instance, the user introduces a constraint to model the hypothesis that the Calcium Carbonate administration happens between 12pm and 2pm, which is $[12 \leq CC - RT \leq 14]_{HYP}$. On the other hand, in our approach we consider *implicit* hypothetical constraints, tagged with the label "REF", in two different circumstances:

(i)   When we look at the time of execution of future actions in order to have (or not have) an interaction (facilities 3 and 4), the technical procedure we adopt (see "*Solution Extraction*" in Section 5.4) involves the assertion of the hypothetical temporal constraint that the interaction occurs or not, i.e., that the effects of the actions intersect or do not intersect in time.

(ii) When we support the refinement of solutions (see the *"Solution Refinement"* in Section 5.4), each refinement is coped with by our approach as a hypothetical constraint to be added.

However, STP (see Appendix 1 for further details) does not directly support hypothetical reasoning. As a first approximation, hypothetical reasoning can be simply provided by:

(1) Generating a copy MN' of the minimal network.
(2) Adding the hypothetical constraint(s) to MN'.
(3) Performing the propagation on MN' using Floyd-Warshall's algorithm and obtaining a new minimal network MN".
(4) Inspecting the new minimal network MN".

The resulting minimal network MN" is the output for the user, and can be removed later on. However, when adding a new constraint on some future actions, a hidden problem must also be taken into account: the addition can make *some* or even *all* the paths of the CIGs containing such actions temporally inconsistent and, thus, not *executable*. Of course, the **executability** of future paths must be detected by the temporal reasoner, and signaled to the user. In order to do so, each alternative path in the CIGs stemming from the actions involved in a hypothetical constraint must be considered separately. There are two cases. In the simpler case the hypotheses involve the actions of one of the two CIGs only. In such a case, for each one of the possible future paths, a temporary STP containing the hypothesized constraints and all the constraints in the path must be generated, and temporal reasoning must be executed on each temporary STP to check its consistency. In the more complex case the hypothesized constraints relate actions of two CIGs. In such a case, future paths must be checked pairwise (i.e., one future path from the first CIG and one from the second one). In both cases, if some (or all) the temporary STPs are inconsistent, a warning is sent to the user. Finally, the different temporary STPs can be deleted.

Example 1 (Hypothetical reasoning). In Example 1 (query Q2), the hypothesis that the first Erythromycin administration E1 happens the same day of the second Warfarin administration W2 (constraint $[0 \leq E1_P - W2_P \leq 0]_{HYP}$) involves actions belonging to the two CIGs, thus all the pairwise combinations of the future paths must be considered. However, only one future path is possible in each one of the two CIGs, the one containing the execution of W3 for the first CIG and the one containing the execution of E2 for the second one. Thus, there is only one combination of the future paths of the two CIGs, and only one temporary STP, containing W3 and E2, is built to check whether the CIGs are still executable after the addition of the hypothesis. In Table 4, we show the minimal network resulting from the propagation of the temporary STP. Since it is consistent, the hypothesis can be added to the original STP and then the temporary STP can be discarded.   ∎

	RT	W1ₚ	W2ₚ	W3ₚ	WTₛ	WTₑ	DECₚ	ETₛ	E1ₚ	E2ₚ	ETₑ
RT	0	0	1	2	0	∞	0	1	1	2	∞
W1ₚ	0	0	1	2	0	∞	0	1	1	2	∞
W2ₚ	-1	-1	0	1	-1	∞	-1	0	0	1	∞
W3ₚ	-2	-2	-1	0	-2	∞	-2	-1	-1	0	∞
WTₛ	0	0	1	2	0	∞	0	1	1	2	∞
WTₑ	-2	-2	-1	0	-2	0	-2	-1	-1	0	∞
DECₚ	0	0	1	2	0	∞	0	1	1	2	∞
ETₛ	0	0	1	2	0	∞	0	0	1	2	∞
E1ₚ	-1	-1	0	1	-1	∞	-1	0	0	1	∞
E2ₚ	-2	-2	-1	0	-2	∞	-2	-1	-1	0	∞
ETₑ	-2	-2	-1	0	-2	∞	-2	-1	-1	0	0

Table 4. Minimal network built to check hypothesis consistency for query Q2.

5.3.4 Non restriction check

A significant extension is required in order to deal with the combination of explicit and/or implicit hypothetical constraints and constraints regarding imprecise past or ontological constraints.

In the following, we first motivate the extension and then we describe how we have achieved it. First of all, we need to distinguish among "basic" and "hypothetical" constraints. In our approach, we consider three sources of "basic" constraints: (i) the constraints in the log (LOG label), (ii) the constraints in the CIG (CIG label) and (iii) the constraints in the ontological knowledge (ONTO label). Such constraints must be conjunctively consistent (in particular, we assume that each log – i.e., each CIG execution – is conformant with the CIG) and their propagation through Floyd-Warshall's algorithm results in the minimal network of "basic" constraints (henceforth: *basic_mn*).

The cue point is to realize that not all the "basic" constraints in *basic_mn* have the same nature. On the one hand, the constraints regarding the future, not yet performed, actions in the CIGs are "*refinable*", in the sense that the user-physicians executing the CIGs may restrict them, within the range of possibilities admitted by the constraints. On the other hand, the constraints in the ontology (ONTO constraints) concern the delays between causes and effects of actions, and are not under the control of physicians. Analogously, also the constraints concerning the time of execution of actions in the log (LOG constraints) are not refinable, since such actions have been already executed and their execution time, although possibly temporally imprecise, is out of anyone's control. As a simple example, suppose that the minimal network concerning an imprecise log along with its corresponding CIGs and ontological knowledge contains the constraint that A started between 10 and 20 (i.e., $10 \leq A_S - RT \leq 20$). This means that A_S has occurred at a specific time t_{A_S} that may be any time between 10 and 20. One cannot restrict the given bounds, since there is the risk that, in such a way, one loses the actual time of occurrence.

Whenever some hypothetical constraints have to be managed, such additional constraints must be propagated together with the "basic" ones (i.e., considering *basic_mn*). However, for the reasons discussed above, we want to avoid that the LOG and ONTO constraints are restricted by the addition of hypotheses.

Technically speaking, this means that the LOG and ONTO constraints in *basic_mn* must be contained into the corresponding constraints in the minimal network considering also the "additional" ones (*all_mn*). Notably, in STP the check of containment of a set of constraints can be performed by considering separately each constraint in the set.

In particular, when some hypotheses have to be managed, we resolve two STPs: the first one (*all_stp*) is the "normal" STP built as described in Section 5.2 and considering all types of constraints, while the second (*basic_stp*) contains only the "basic" constraints (i.e., the ones tagged with labels ONTO, CIG and LOG). Technically speaking, *basic_stp* is obtained removing from *all_stp* all the constraints labelled with HYP and REF. Floyd-Warshall's constraint propagation is then performed on both STPs, obtaining two minimal networks: *all_mn* and *basic_mn*. Then, for each pair of points *x* and *y* temporally constrained in LOG or ONTO constraints, the algorithm checks whether all the LOG or ONTO constraints regarding *x* and *y* in *basic_mn* are contained into the constraints between *x* and *y* in *all_mn*. Besides *all_mn*, the algorithm *NonRestrictionCheck* also returns a Boolean value with the result of the containment check. For the sake of simplicity, in the algorithm we treat labels as sets of constraints; thus, *all_stp* – *(HYP ∪ REF)* means "the set of constraints obtained removing from *all_stp* all the constraints labeled with HYP or REF".

```
NonRestrictionCheck algorithm
Input: basic_stp
Output: ‹boolean, set‹Constraint››

begin
 basic_stp ← all_stp - (HYP ∪ REF);
 all_mn ← Floyd-Warshall(all_stp);
 basic_mn ← Floyd-Warshall(basic_stp)
 foreach c ≤ x - y ≤ d ∈ (LOG ∪ ONTO) do
 begin
   let c₁ ≤ x - y ≤ d₁ ∈ basic_mn
   let c₂ ≤ x - y ≤ d₂ ∈ all_mn
   if [c₁,d₁] ⊈ [c₂,d₂] return ‹false, all_mn›
 end
 return ‹true, all_mn›
end
```

Example 1 (Non restriction). To answer, for instance, query Q2 above, the hypothetical constraint $[0 \leq E1_P - W2_P \leq 0]_{HYP}$ has to be added to the original STP. For the sake of compactness and readability, the minimal networks *all_mn* and *basic_mn* resulting by the reasoning process are reported as additional material in Table A1 of Appendix 2. After the propagation, LOG and ONTO constraints (represented by bold elements in the table) are the same in both the minimal networks. This ensures that constraints that are not under the control of physicians have been restricted by the reasoning process. ∎

5.4 Result interpretation

As previously discussed, the result of the temporal reasoning algorithms described in Section 5.3 is a minimal network. Obviously, such an output is not easy to consult for user-physicians and does not explicitly show the answer to the given query (see the facilities in Section 4.2). Following the philosophy underlying our previous works in this field, we thus allow users to interact with the system in a natural and user-friendly way in order to obtain an easy-to-interpret result.

We organize the facilities described in Section 4.2 according to the type of answer they provide. Facilities 1 and 2 (*Interaction*) basically query the system to determine whether an interaction can occur. They provide a YES/NO answer (or MAYBE for cases in which a certain result cannot be ascertained). On the other hand, Facilities 3 and 4 (*Time of future actions to avoid (or to have) an interaction*) provide the scenarios (if they exist) satisfying the constraints. This last case can be further refined supporting the user to interactively "select" a scenario among the possible ones. In this section, we explain how such facilities are obtained starting from the minimal network resulting from the extended reasoning described in Section 5.3. However, it is important to stress that, in our approach, the user does not interact directly with the minimal network, but s/he is supported by a graphical interface (see Section 6).

Solution existence verification

For Facilities 1 and 2, we need to check whether the times of the variations (which are the effects/intentions of the actions in two CIGs) intersect in time. In Table 5, we named the checking function *Sol Exist*. Considering two variations A and B, the answer YES must be provided just in case A and B necessarily intersect in time. Such a test can be directly performed through an inspection of the minimal network. Specifically, to ascertain the intersection we have to check whether in the minimal network the following condition is necessarily true:

$$(A_S - B_S \leq 0 \land B_S - A_E \leq 0) \lor (B_S - A_S \leq 0 \land A_S - B_E \leq 0) \qquad \text{(Condition 1)}$$

where A_S (B_S) and A_E (B_E) indicate respectively the starting and ending points of A (B). On the other hand, the non-intersection is ascertained by checking the necessity of the following condition:

$$(A_E - B_S < 0) \lor (B_E - A_S < 0) \qquad \text{(Condition 2)}$$

Notice that, since Condition (1) is the negation of Condition (2), we can easily verify the necessity of one of the two conditions by verifying the non-possibility of the other one and vice versa.

Example 1 (Solution Existence Verification). The minimal network resulting from Example 1 to answer query Q1 is shown in Table A2 (see Appendix 2).

To ensure the interaction, the intervals $[AN2_S, AN2_E]$ and $[RM1_S, RM1_E]$ must overlap, thus the Condition (1) must hold, i.e.:

$$(AN2_S - RM1_S \leq 0 \wedge RM1_S - AN2_E \leq 0) \vee (RM1_S - AN2_S \leq 0 \wedge AN2_S - RM1_E \leq 0)$$

To ensure the non-interaction, the Condition (2) must hold, i.e.:

$$(AN2_E - RM1_S < 0) \vee (RM1_E - AN2_S < 0)$$

Given the minimal network, both the conditions are possible, but not necessary. Then, the answer of the system is MAYBE.

On the other hand, in the minimal network evaluated to answer query Q2 (see Table A1) the necessity of Condition (1) is verified (i.e., Condition (2) is not possible). Then the answer is YES, i.e., under the given hypothesis, the two actions certainly interact in time. ∎

Solution extraction

Facilities 3 and 4 require finding one or more configurations for future actions such that the given interaction certainly occurs or certainly does not occur (*Get Sol* in Table 5).

Given the minimal network that is the output of the extended STP reasoning, the goal of the *Get Sol* procedure is to return a minimal network representing the configurations of the future actions for which the interaction certainly occurs or certainly does not occur. Technically speaking, given the intervals of occurrence $A = [A_S, A_E]$, $B = [B_S, B_E]$ of two interacting variations and the minimal network, the procedure has to obtain a "restricted" minimal network in which Condition (1) (to have the interaction) or Condition (2) (to avoid the interaction) are *necessary*.

However, the constraints regarding the delay between the actions and their intentions/effects and the durations of the latter are not under the control of physicians and at execution time they can assume each value admitted by the constraints. Thus, they cannot be restricted by the addition of Condition (1) or Condition (2) to the minimal network.

Our idea to manage such a problem is simple: we "project" the constraints imposing that A and B overlap (or do not overlap) back to the corresponding constraints between the time of execution of the actions causing them (which, on the other hand, are under the control of users), and then we propagate in order to find the strictest constraints (minimal network) or an inconsistency.

In particular, in STP, each event E has a certain (say Cert(E)) and a possible (say Poss(E)) extension with respect to a reference time. The two variations A and B necessarily overlap if Cert(A) ∩ Cert(B) ≠ ∅, while they necessarily do not overlap if Poss(A) ∩ Poss(B) = ∅. Given the minimal network represented by a matrix D (see Appendix 1), these conditions map onto the Conditions (3) and (4) holding between the time points Act_A, i.e., the ending point of the action causing A, and Act_B, i.e., the ending point of the action causing B:

$$-D[B_E, Act_B] - D[Act_A, A_S] \leq Act_A - Act_B \leq -D[A_E, Act_A] - D[Act_B, B_S]$$

$$\text{(Condition 3)}$$

$$Act_A - Act_B < -D[B_S, Act_B] - D[Act_A, A_E] \vee$$
$$Act_A - Act_B > D[Act_B, B_E] + D[A_S, Act_A]$$

$$\text{(Condition 4)}$$

Thus, adding such conditions to the minimal network *mn* and propagating the constraints results in the minimal network in which the original conditions of interaction or of non-interaction are *necessary*, independently of the actual values assumed by the starting and ending points of A and B at execution time. Since STP does not support disjunctions, to cope with Condition (4) we add each disjunct to a copy of the minimal network, obtaining two minimal networks (or one, in case one of the two minimal networks results inconsistent) in which the condition is satisfied.

Solution refinement. The obtained minimal networks are an implicit representation of the scenarios of future action executions in which the interaction occurs (or is avoided). We propose an iterative and interactive process to refine such an initial solution. Here, we abstractly propose such a process. In Section 6, describing the graphical interface to support it, we will show in detail how it works.

The resulting minimal networks contain constraints between any pair of time points. In such a form, constraints are not easily understandable for users. In order to facilitate their understanding and to support a graphical representation, we "align" the times of the actions and of the variations with

respect to a specific point in time. At the beginning of the process, the user chooses an *aligning point* AP (by default, AP is the reference time RT) among the ones already present in the minimal network. Given AP, the system returns to the user all the constraints between AP and the endpoints of the times of future actions. Such constraints represent the *intervals of existence* with respect to AP of the endpoints of the future actions, i.e., their certain and possible times. Then, iteratively, the system allows the user to add a refinement constraint (tagged with the label REF) that restricts an interval of existence. The refinement constraint is treated as a hypothetical temporal constraint. Thus, the system propagates such a refinement constraint using the previously described algorithms and, in case of imprecise past, checking the non-restriction of LOG and CIG constraints. Then it returns a new scenario consistent with the refinement as well as the information on whether the addition of the new constraints still preserve the executability of at least one of the alternative future paths starting from the given interaction. The user can analyze the new scenario and further modify/restrict it or decide that one or more intervals of existence are definitely fixed. The process ends when the user has fixed all the intervals of existence. Notably, the user can exploit the above process to refine each interval of existence reducing it to a specific point in time or advocate a special option of the system that identifies a specific solution on the basis of the chosen intervals of existence. Such an option can be computed through the application of [39].

Example 2 (Solution Extraction and Refinement). As an example, we consider the solution extraction and refinement for Example 2. The minimal network obtained from the extended temporal reasoning to answer Query Q3 is shown in Table A3 in Appendix 2.

After that, to answer the query, we need to add one of the two elements of the disjunction in Condition (4) to the minimal network. In the example, with $Act_A = CC_P$ and $Act_B = NA_P$, the first element of the condition results in $CC_P - NA_P < -(-1) - 6 \Rightarrow CC_P - NA_P < -5$. Adding such a constraint to the minimal network and propagating it, we obtain the minimal network shown in Table A4 in Appendix 2.

Now, we suppose that the physician is interested in extracting a solution in which the Calcium Carbonate is administered at 2pm and s/he wants to prescribe the Nalidixic Acid administration to happen no later than 9pm. Then the system adds to the minimal network of Table A4 the refinement constraints $[14 \leq CC_P - RT \leq 14]_{REF}$ and $[NA_P - RT \leq 21]_{REF}$. The result of the propagation is shown in Table A5 in Appendix 2. The solutions regarding the Nalidixic Acid administration of such minimal network are $\{CC_P = 14 \land NA = 20\}$ and $\{CC_P = 14 \land NA = 21\}$, which correspond to the time of execution of such an action that, given the physician's refinements, certainly avoids the interaction . ■

5.5 Supporting the facilities

Until now, we have defined the "basic capabilities" that are required in order to support facilities (1)-(4) in each one of the four different scenarios discussed in Section 4.2. We can finally "wrap up", showing how such basic capabilities can be combined in order to cope with each <facility, scenario> pair. Results are graphically summarized in Table 5, where we list the reasoning and interpretation capabilities required in order to provide each facility in each scenario.

First, as previously discussed, the procedure for the interpretation of the result depends on the selected facility. In particular, Facilities 1 and 2 require the application of *Sol Exist*, while Facilities 3 and 4 require the application of *Get Sol*. On the other hand, STP reasoning extensions are treated as "basic reasoning elements", which can be combined to support a specific facility.

The *Partonomic* reasoning is exploited in each scenario, since it manages the constraints deriving from partonomic relations, which is an intrinsic characteristic of the most CIG formalisms, such as GLARE. The *Class-instance* reasoning is applied when a temporal log is available, managing the relation between the temporal constraints regarding the "classes" of actions in the CIGs and possibly in the ontological knowledge, and the instances contained in the execution log. In addition, in the cases in which the log is temporally imprecise, also the *Non-restriction check* procedure is performed. *Hypothetical* reasoning is performed whenever some hypothetical constraint is explicitly (Facilities 2 and 4) or implicitly (Facility 3) considered. In all such cases, both the executability of future actions and the check of non-restriction of past actions (i.e., of the constraints in the log) and of the ontological constraints is taken into account (see Section 5.4).

It is worth noticing that Facility 1 (*Interaction*) cannot be managed (N/A in Table 5) in case of absence of a temporal log. Indeed, in such a situation, one is considering two CIGs abstractly, i.e., without considering any patient, and without hypothesizing any temporal alignment between them.

In such a context, the times of execution of the two CIGs are totally unknown and unrelated, so that no form of temporal analysis can be supported.

Facility / *Scenario*	*1* *Interaction*	*2* *Interaction* *(what-if)*	*3* *Time of future actions to avoid (or to have) an interaction*	*4* *Time of future actions to avoid (or to have) an interaction (what-if)*
Non-temporal log/ No log	N/A	Partonomic Hypothetical Non Restriction Sol Exist	Partonomic Hypothetical Non Restriction Get Sol	Partonomic Hypothetical Non Restriction Get Sol
Temporally exact log	Partonomic Class-instance Sol Exist	Partonomic Class-instance Hypothetical Non Restriction Sol Exist	Partonomic Class-instance Hypothetical Non Restriction Get Sol	Partonomic Class-instance Hypothetical Non Restriction Get Sol
Temporally imprecise log	Partonomic Class-instance Sol Exist	Partonomic Class-instance Hypothetical Non Restriction Sol Exist	Partonomic Class-instance Hypothetical Non Restriction Get Sol	Partonomic Class-instance Hypothetical Non Restriction Get Sol

Table 5. Rows represent the different log scenarios and columns the requested facility. In each cell of the table, the upper part lists the algorithms used for the temporal reasoning, while the lower (underlined) element explains which methodology is used to interpret the result. N/A stands for "not available".

6 Visualization and user interaction

We further enhance our Mixed-Initiative approach by providing a graphical interface for the visualization and refinement of the results, providing user-physicians with a friendly interface to cope with temporal constraints at a high level of abstraction and without having to directly interact with STP constraints.

We provide a basic interface for the visualization of the result of the procedure *Sol Exist*, and a richer interface for the interaction between the user physician and the system in the case of the procedure *Get Sol*.

In the case of *Sol Exist*, our goal is to show to the physician how the interacting variations overlap (or do not overlap) in time. Thus, the system represents in two aligned timelines (as discussed in Section 5.4, the user can choose an aligning point AP) the intervals of existence of the interacting variations. In each timeline, two rectangles represent the intervals of existence of the endpoints of the respective variation: the upper rectangle represents the time interval in which the variation can start, while the lower one represents the time interval in which the variation can end. On the other hand, the time interval, if any, included between the two rectangles (represented with a colored line) represents the time when the variation certainly holds.

Example 1 (Visualization). In Figure 8, we show the graphical representation returned by our system to support the "YES" answer to query Q2, i.e., the interaction certainly occurs in time. The upper timeline (red rectangles) represents the interval of existence of the Anticoagulation effect (considering Table A1 in Appendix 2) caused by the Warfarin administration W2, while the lower timeline (green rectangles) represents the effect Reducing metabolism of the Erythromycin administration E1. The intervals are drawn considering the reference time RT as aligning point, i.e., AP = RT. It is possible to observe that the certain parts of the two variations necessarily overlap in time since both the effects certainly hold in hour 2. ∎
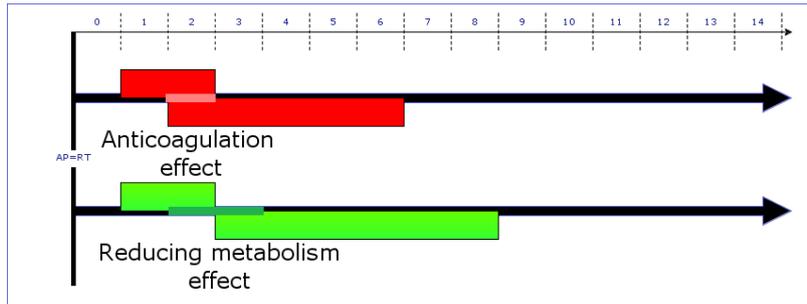
Figure 8. Graphical representation of the result of the analysis of interaction between the "*Anticoagulant*" effect of the warfarin administration and the "*Reducing metabolism*" effect of the erythromycin.

In the case of *Get Sol*, the graphical representation described above is extended with a list of timelines, each of them representing the intervals of existence of the endpoints of a future action (see Figure 9). The user can drag the left and right borders of the rectangles restricting the intervals of existence of the endpoints. Each time a rectangle is resized, the corresponding REF constraint (see Section 5.4 above) is added to the underlying STP and it is propagated. After the propagation, the interface is updated according to the resulting minimal network. This allows a user physician to directly observe the result of the refinement s/he has just added in terms of (i) restrictions for the times of execution of other actions and (ii) effects on the overlapping of the interacting variations. By default, refinements are provisional: if the user decides to enlarge an interval of existence (within the limits imposed by the original minimal network), all the previously added refinements are removed. To prevent REF constraints to be removed, physician can fix them, adding them permanently to the STP (the relative rectangle is filled with grey color and is no resizable by the user). As previously discussed, the refinement process ends when all the intervals of existence are fixed or when the user requires a specific solution on the basis of the chosen intervals.

Example 2 (Visualization). In Figure 9, we show a step of the solution extraction for Example 2. In particular, the first two timelines represent the intervals of existence of the interacting effects Urine Alkalinization and Nalidixic Acid absorption, while the third and the fourth ones represent the intervals in which the Calcium Carbonate and the Nalidixic Acid administrations can be executed (points CC_P and NA_P). The user can restrict these two intervals, which equals to add REF constraints to the underlying minimal network. ■
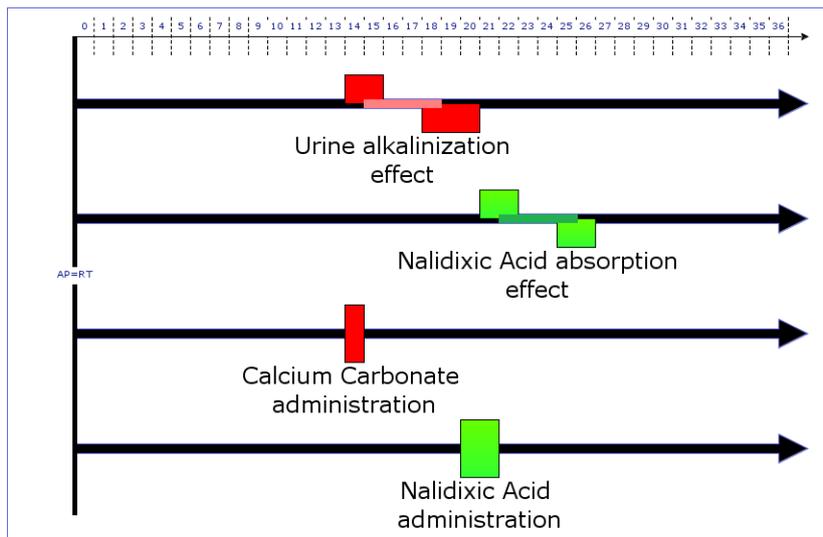

Figure 9. Graphical interface for the result of the analysis of the interaction between the Calcium Carbonate and the Nalidixic Acid administration.

7 Discussion and conclusions

The detection and management of the interactions between CIGs in the treatment of comorbid patient is a "hot topic" in the area of Medical Informatics. In this paper, we presented the *first* approach that deeply takes into account the *temporal* dimension in the treatment of comorbid patients, both considering analytical aspects, i.e., temporal analysis to detect interactions, and decision-making features, i.e., facilities to obtain the times of future actions to have/avoid interactions. This is, in our opinion, a crucial advance with respect to the state of the art: a non-temporal analysis can only consider *theoretically possible* interactions between actions in different CIGs (e.g., conflicts between their goals or effects), while *actual* interactions occur *in time*, i.e., just in case that the considered goals or effects *overlap in time*. In this paper, we have proposed an innovative **domain-independent** approach to **temporal detection and analysis of interactions between guidelines** considering different sources of temporal information (CIGs, general medical knowledge, and execution logs) and proposing a wide set of user-friendly facilities, covering four different application scenarios.

In Section 2.1, we have analysed the state of the art of CIG approaches coping with comorbidities. The approach we have presented in this paper clearly differs from all of them, since it is the first one that takes into account time in the detection of interactions between CIGs. This is a clear advance in the state of the art. To achieve such a main result, we

(i)   Have proposed an enhanced representation of actions, effects, intentions, and interactions, enriched with temporal notions such as the time intervals of occurrence, and the temporal constraints between them.

(ii)   Have developed advanced temporal reasoning techniques which extend the basic STP framework in order to provide different types of temporal facilities, needed to cope with temporal detection in different possible scenarios.

Considering **_knowledge representation_**, our ontology of actions, intentions and interactions is quite similar to the conceptual model in [16]. Such an approach, however, does not cope with time and adopts an ad hoc formalism for CIG representation. Our temporal ontology is mainly derived by our previous representation of time, action-types and causation in [35]. Furthermore, there are interesting analogies with ASBRU temporal ontology [14], which also models action intentions, and Allen's temporal constraints [34], as well as delays and durations. On the other hand, we explicitly model also "mixed" qualitative relations between time intervals and time points, identified in [33], but not considered in ASBRU. However, ASBRU did not consider action interactions and, thus, it is not geared toward the treatment of comorbidities.

In general, temporal issues are pervasive in the CIG context, and many previous approaches have faced some of them (see, e.g., the survey in [24]), mostly from the point of view of the representation of temporal knowledge. On the other hand, few CIG approaches have faced the problem of **_reasoning about temporal constraints_** between actions, which is the core contribution of this paper [25,26,40]. However, none of such approaches specifically focused on the temporal phenomena related to the temporal detection of interactions between CIG actions. As a consequence, none of them focused on the problems we tackled in Section 5 and, in particular, with partonomic, class-instance and hypothetical temporal reasoning, or on the non-restriction check (see Section 5.3), which are among the major original contributions of the work in this paper.

Regarding **_comorbidity management_**, besides the temporal aspects already commented in this section, another important distinguishing aspect must be highlighted. In Section 2.2, we identified three different stages in the merging of clinical guidelines: the detection of interactions between CIGs, their solving and the final merge of the CIGs into a unique treatment. Most of the approaches in literature implicitly or explicitly focus on the third stage: given the CIGs, a set definitions of possible interactions and a set of managements for such interactions, they automatically return as output possible treatments. Our approach and the one in [16] are, to the best of our knowledge, the only ones that provide support also to the *detection* of interactions by reasoning over models of *general medical knowledge*. In addition, our approach is the only one that performs such a task in a mixed-initiative way, providing a tool that supports and encourages a collaboration between the human user and the automatic system, allowing users to analyse interactions at different levels of abstraction.

A preliminary and short version (11 pages) of this paper has been published in the proceedings of a recent HEALTHINF conference [10]. The general ideas underlying our temporal approach, the basic temporal ontology and the identification of the different scenarios and facilities have been briefly discussed in such a paper. This paper not only extends and systematizes such preliminary discussions, but also substantiates them by proposing (in Sections 4, 5 and 6, which are the core of this paper and are almost entirely new) detailed algorithms for the provided analysis options. Notably, in this paper we have also covered all of the facilities outlined in [10], where the cases of hypothetical queries with temporally imprecise log have been left as open problems to be addressed in future work.

We have implemented a prototypical version of our approach and we are currently running an experimental evaluation of it. The kind of evaluation we are carrying on with expert physicians is driven by the fact that our approach and methodology are *domain-independent*. As a consequence, in our current experiments, our main goal is not the development of a specific instantiation of our approach, building a "complete" model covering a specific case of comorbidity – consider, e.g., the treatment of the interactions between hypertension, diabetes mellitus and heart failure presented in [19]. On the contrary, we aim at showing the generality of our domain-independent approach. For such a reason, we considered different (but specific, and thus nor very large) examples, taken from different domains, trying to cover as many different types of interactions as possible. A panorama of the different cases we considered until now can be found in [41], but, of course, more extensive experimentations are still needed. In particular, as a specific example, we have devoted attention to the treatment of the interactions between the treatments for venous thromboembolism and respiratory tract infections, and between venous thromboembolism and peptic ulcer, considering the guidelines provided by the British National Institute for Health and Care Excellence (www.nice.org.uk), and some of the interactions identified in [42–45].

For each experiment, the goal was not to show the completeness of our model, but to test whether, on specific examples in different application domains, our approach provides or not physicians with an adequate support to detect temporal interactions between multiple guidelines. Each experiment has been carried out in four steps:

(i)   First, the proper pieces of CIGs have been acquired, considering the guidelines in the medical literature, and taking advantage of the advice of expert physicians. Specific attention has been devoted to the identification and representation of the temporal constraints in the guidelines. Such an acquisition has been done taking advantage of GLARE acquisition tool.

(ii)   Second, the medical knowledge about the interactions has been acquired on the basis of the medical literature and with the advice of expert physicians. We used Protégé [37] in order to facilitate the acquisition of such a knowledge.

(iii)   The expert physicians have designed some "typical" patient, and a partial log to represent her\his treatment up to the time when the analysis of interaction is carried on.

(iv)   The expert physicians have used a prototypical implementation of our approach in order to temporally analyse the potential interactions on the CIGs for the given patient on the basis of the current patient's log.

The results of our experiments have been quite satisfactory, in the sense that most physicians have appreciated the support provided by our approach to temporally analyse interactions (step (iv)). Indeed, the temporal analysis necessarily involves the "merge" and propagation of different temporal constraints emerging from different knowledge sources. Temporal constraint propagation is not at all a task that physicians – and humans in general – consider "natural" or "easy", so that most of them positively accepted the support of a system helping them in such a complex task. Indeed, in complex cases, it was not possible to compare the results of the system with respect to the expectations of expert physicians, for the simple reason that physicians (but, more generally, humans) have not the capability of performing complex propagations of constraints, so that, basically they could not even formulate approximate expectations. Some comparisons were only possible considering "simple" problems, in which only a very limited amount of constraint propagation had to be performed. In such cases, our approach usually has a "pessimistic" behaviour, meaning that, for instance, in the case of Facilities 3 and 4 to avoid an interaction our temporal suggestions tend to be stricter (more constrained) than the ones provided by expert physicians or by the medical literature. This characteristic is motivated in part by the fact that medical reasoning

involves, besides temporal reasoning, also probabilistic reasoning (e.g., physicians consider two actions as non-interacting if the probability that they interact is less than a certain threshold), while constraint propagation provides as output strict temporal constraints that ensure that no interaction at all can occur.

However, we are aware that more substantial experimental evaluations are needed. In particular, as our future work, we aim not only at investigating further cases of interactions, in different domains, but also to work "in depth" on a specific case of comorbidity, to devise a specific and as complete as possible instantiation of our approach to manage it. Defining a complete instantiation of our approach on a specific example can also allow us to apply more thorough evaluation measures for the knowledge represented in our model, such as the ones provided by [46].

Acknowledgments

References

[1]   Institute of Medicine, Committee on Quality Health Care in America, Crossing the Quality Chasm: A New Health System for the 21st Century, (2001). https://www.iom.edu:443/Reports/2001/Crossing-the-Quality-Chasm-A-New-Health-System-for-the-21st-Century.aspx (accessed March 10, 2015).

[2]   Guidelines International Network, Guidelines International Network Website. (n.d.). http://www.g-i-n.net/ (accessed October 14, 2014).

[3]   C. Gordon, J.P. Christensen, eds., Health telematics for clinical guidelines and protocols, IOS Press, Amsterdam, Netherlands, 1995.

[4]   D.B. Fridsma, Special Issue on Workflow Management and Clinical Guidelines, Journal of the American Medical Informatics Association. 22 (2001) 1–80.

[5]   A. Ten Teije, S. Miksch, P. Lucas, eds., Computer-based medical guidelines and protocols: a primer and current trends, IOS Press, Amsterdam, 2008. http://dl.acm.org/citation.cfm?id=1479604.

[6]   M. Peleg, Computer-interpretable clinical guidelines: A methodological review, Journal of Biomedical Informatics. 46 (2013) 744–763. doi:10.1016/j.jbi.2013.06.009.

[7]   D. Riaño, A. Collado, Model-Based Combination of Treatments for the Management of Chronic Comorbid Patients, in: Artificial Intelligence in Medicine, Springer, 2013: pp. 11–16.

[8]   M. Michalowski, S. Wilk, W. Michalowski, D. Lin, K. Farion, S. Mohapatra, Using Constraint Logic Programming to Implement Iterative Actions and Numerical Measures during Mitigation of Concurrently Applied Clinical Practice Guidelines, in: Proceedings of AIME, Springer Berlin Heidelberg, 2013: pp. 17–22.

[9]   L. Piovesan, G. Molino, P. Terenziani, An ontological knowledge and multiple abstraction level decision support system in healthcare, Decision Analytics. 1 (2014) 1–24. doi:10.1186/2193-8636-1-8.

[10]   L. Piovesan, L. Anselma, P. Terenziani, Temporal detection of guideline interactions, in: Proceedings of the International Conference on Health Informatics (HEALTHINF-2015), Scitepress, 2015: pp. 40–50. doi:10.5220/0005186300400050.

[11]   R.I. Sato, D.R. Gray, S.E. Brown, Warfarin interaction with erythromycin, Arch. Intern. Med. 144 (1984) 2413–2414.

[12]   P. Terenziani, G. Molino, M. Torchio, A modular approach for representing and executing clinical guidelines, Artificial Intelligence in Medicine. 23 (2001) 249–276. doi:10.1016/S0933-3657(01)00087-2.

[13]   P. Terenziani, S. Montani, A. Bottrighi, G. Molino, M. Torchio, Applying artificial intelligence to clinical guidelines: the GLARE approach, Stud Health Technol Inform. 139 (2008) 273–282.

[14]   Y. Shahar, S. Miksch, P. Johnson, The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines, Artif Intell Med. 14 (1998) 29–51.

[15] R. Dechter, I. Meiri, J. Pearl, Temporal Constraint Networks, Artif. Intell. 49 (1991) 61–95. doi:10.1016/0004-3702(91)90006-6.

[16] V. Zamborlini, R. Hoekstra, M.D. Silveira, C. Pruski, A. ten Teije, F. van Harmelen, A Conceptual Model for Detecting Interactions among Medical Recommendations in Clinical Guidelines - A Case-Study on Multimorbidity, in: K. Janowicz, S. Schlobach, P. Lambrix, E. Hyvönen (Eds.), Knowledge Engineering and Knowledge Management - 19th International Conference, EKAW 2014, Linköping, Sweden, November 24-28, 2014. Proceedings, Springer, 2014: pp. 591–606. doi:10.1007/978-3-319-13704-9_44.

[17] S. Wilk, W. Michalowski, M. Michalowski, K. Farion, M.M. Hing, S. Mohapatra, Mitigation of adverse interactions in pairs of clinical practice guidelines using constraint logic programming, J Biomed Inform. 46 (2013) 341–353. doi:10.1016/j.jbi.2013.01.002.

[18] I. Sánchez-Garzón, J. Fernández-Olivares, E. Onaindia, G. Milla, J. Jordán, P. Castejón, A Multi-agent Planning Approach for the Generation of Personalized Treatment Plans of Comorbid Patients, Artificial Intelligence in Medicine. (2013) 23–27. doi:10.1007/978-3-642-38326-7_4.

[19] J.A. López-Vallverdú, D. Riaño, A. Collado, Rule-Based Combination of Comorbid Treatments for Chronic Diseases Applied to Hypertension, Diabetes Mellitus and Heart Failure, in: R. Lenz, S. Miksch, M. Peleg, M. Reichert, D. Riaño, A. ten Teije (Eds.), Process Support and Knowledge Representation in Health Care, Springer Berlin Heidelberg, 2013: pp. 30–41.

[20] B. Jafarpour, S.S.R. Abidi, Merging Disease-Specific Clinical Guidelines to Handle Comorbidities in a Clinical Decision Support Setting, in: Artificial Intelligence in Medicine, 2013: pp. 28–32.

[21] E. Merhej, S. Schockaert, T.G. McKelvey, M. De Cock, Generating conflict-free treatments for patients with comorbidity using ASP, in: 8th International Workshop on Knowledge Representation for Health Care (KR4HC'16), 2016: pp. 93--100.

[22] Y. Zhang, Z. Zhang, Preliminary Result on Finding Treatments for Patients with Comorbidity, in: S. Miksch, D. Riaño, A. ten Teije (Eds.), Knowledge Representation for Health Care, Springer International Publishing, 2014: pp. 14–28.

[23] P. Fraccaro, M. Arguello Castelerio, J. Ainsworth, I. Buchan, Adoption of Clinical Decision Support in Multimorbidity: A Systematic Review, JMIR Medical Informatics. 3 (2015) e4. doi:10.2196/medinform.3503.

[24] P. Terenziani, E. German, Y. Shahar, The temporal aspects of clinical guidelines, Stud Health Technol Inform. 139 (2008) 81–100.

[25] L. Anselma, P. Terenziani, S. Montani, A. Bottrighi, Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines, Artificial Intelligence in Medicine. 38 (2006) 171–195. doi:10.1016/j.artmed.2006.03.007.

[26] C. Combi, B. Oliboni, A. Gabrieli, Conceptual Modeling of Clinical Pathways: Making Data and Processes Connected, in: J.H. Holmes, R. Bellazzi, L. Sacchi, N. Peek (Eds.), Artificial Intelligence in Medicine - 15th Conference on Artificial Intelligence in Medicine, AIME 2015, Pavia, Italy, June 17-20, 2015. Proceedings, Springer, 2015: pp. 57–62. doi:10.1007/978-3-319-19551-3_7.

[27] E. Horvitz, Uncertainty, Action, and Interaction: In Pursuit of Mixed-Initiative Computing, IEEE Intelligent Systems. 14 (1999) 17–20.

[28] International Health Terminology Standards Development Organisation, SNOMED Clinical Terms, (2015). http://www.ihtsdo.org/snomed-ct (accessed January 13, 2017).

[29] WHO Collaborating Centre for Drug Statistics Methodology, Anatomical Therapeutic Chemical classification system, (n.d.). http://www.whocc.no/atc/ (accessed October 14, 2014).

[30] L. Piovesan, G. Molino, P. Terenziani, Supporting Multi-Level User-Driven Detection of Guideline Interactions, in: Proceedings of the International Conference on Health Informatics (HEALTHINF-2015), Scitepress, 2015: pp. 413–422. doi:10.5220/0005217404130422.

[31] L. Piovesan, P. Terenziani, A Mixed-Initiative approach to the conciliation of Clinical Guidelines for comorbid patients, in: Knowledge Representation for Health Care, Springer International Publishing, Pavia, 2015: pp. 95–108.

[32] L. Piovesan, P. Terenziani, A Constraint-Based Approach for the Conciliation of Clinical Guidelines, in: Advances in Artificial Intelligence - IBERAMIA 2016, Springer International Publishing, 2016: pp. 77–88.

[33] M. Vilain, A System for Reasoning About Time, in: D.L. Waltz (Ed.), Proceedings of the National Conference on Artificial Intelligence. Pittsburgh, PA, August 18-20, 1982, AAAI Press, 1982: pp. 197–201. http://www.aaai.org/Library/AAAI/1982/aaai82-047.php (accessed February 25, 2015).

[34] J.F. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM. 26 (1983) 832–843. doi:10.1145/182.358434.

[35] P. Terenziani, P. Torasso, Time, action-types, and causation: an integrated analysis, Computational Intelligence. 11 (1995) 529–552. doi:10.1111/j.1467-8640.1995.tb00047.x.

[36] M. Vilain, H. Kautz, P. van Beek, Constraint propagation algorithms for temporal reasoning: a revised report, in: D.S. Weld, J. de Kleer (Eds.), Readings in Qualitative Reasoning About Physical Systems, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990: pp. 373–381.

[37] M.A. Musen, The Protégé Project: A Look Back and a Look Forward, AI Matters. 1 (2015) 4–12. doi:10.1145/2757001.2757003.

[38] P. Terenziani, L. Anselma, A knowledge server for reasoning about temporal constraints between classes and instances of events, Int. J. Intell. Syst. 19 (2004) 919–947. doi:10.1002/int.20030.

[39] A. Gerevini, M. Cristani, On Finding a Solution in Temporal Constraint Satisfaction Problems, in: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes, Morgan Kaufmann, 1997: pp. 1460–1465.

[40] G. Duftschmid, S. Miksch, W. Gall, Verification of temporal scheduling constraints in clinical practice guidelines, Artificial Intelligence in Medicine. 25 (2002) 93–121. doi:10.1016/S0933-3657(02)00011-8.

[41] L. Piovesan, A Mixed-Initiative Knowledge-Based Decision Support Methodology for the Management of Patients affected by Comorbidities, Ph.D. thesis, University of Turin, 2016.

[42] C.H. Brown, E.A. Natelson, M.W. Bradshaw, P. Alfrey C, T.W. Williams, Study of the effects of ticarcillin on blood coagulation and platelet function, Antimicrob. Agents Chemother. 7 (1975) 652–657.

[43] D. O'Donnell, Antibiotic-induced potentiation of oral anticoagulant agents, Med. J. Aust. 150 (1989) 163–164.

[44] P.J. Rice, R.J. Perry, Z. Afzal, I.H. Stockley, Antibacterial prescribing and warfarin: a review, Br Dent J. 194 (2003) 411–415. doi:10.1038/sj.bdj.4810049.

[45] A.E. Weale, D.J. Warwick, N. Durant, D. Prothero, Is there a clinical interaction between low molecular weight heparin and non-steroidal analgesics after total hip replacement?, Ann R Coll Surg Engl. 77 (1995) 35–37.

[46] E. Shalom, Y. Shahar, M. Taieb-Maimon, G. Bar, A. Yarkoni, O. Young, S.B. Martins, L. Vaszar, M.K. Goldstein, Y. Liel, A. Leibowitz, T. Marom, E. Lunenfeld, A quantitative assessment of a methodology for collaborative specification and evaluation of clinical guidelines, Journal of Biomedical Informatics. 41 (2008) 889–903. doi:10.1016/j.jbi.2008.04.009.

[47] V. Brusoni, L. Console, P. Terenziani, On the computational complexity of querying bounds on differences constraints, Artificial Intelligence. 74 (1995) 367–379. doi:http://dx.doi.org/10.1016/0004-3702(95)00008-3.

Appendix 1. The STP framework

This appendix is intended for readers not familiar with temporal reasoning topics. In this appendix we introduce some preliminaries regarding temporal reasoning and STP [15].

1.1 Representing STPs

STP is based on constraint propagation. An STP constraint is a bound on differences of the form $c \leq x - y \leq d$, where x and y are temporal points and c and d are numbers whose domain can be either discrete or dense. The intuitive temporal interpretation of the constraint is that the *temporal distance* between the time points x and y is between c (minimum distance) and d (maximum distance). It is possible to specify also strict inequalities and $-\infty$ and $+\infty$ can be used to denote infinite lower and upper bounds respectively, i.e., no lower or upper bound. An STP is a set of constraints, i.e. a *conjunction* of STP constraints.

Two representations are often used for STPs: graph and matrix. An STP is represented as a graph whose nodes correspond to the temporal points of the STP and the arcs are labeled with a weight representing the maximum temporal distance between the temporal points. A constraint $c \leq x - y \leq d$ is thus represented by two edges corresponding to the pair of inequalities $x - y \leq d$ and $y - x \leq -c$. For short, usually the arcs are labeled with the interval $[c, d]$. Alternatively, an STP is represented as a matrix D of size $N{\times}N$ where N is the number of temporal points and where the element $D[x, y] = d$ represents the maximum distance d between the points x and y. The minimum distance c is represented as the maximum distance $D[y, x] = -c$ between y and x.

Example. Let us consider the following information concerning temporal points A, B, and C: B occurs between 2 and 4 hours after A, C occurs between 2 and 4 hours after B and between 2 and 6 hours after A. This information can be represented by the following STP S composed by a conjunction of three STP constraints (in this example we assume that the domain is the integers). We provide also the representation of S as a graph and as a matrix.

$$2 \leq B - A \leq 4, \qquad 2 \leq C - B \leq 4, \qquad 2 \leq C - A \leq 6$$
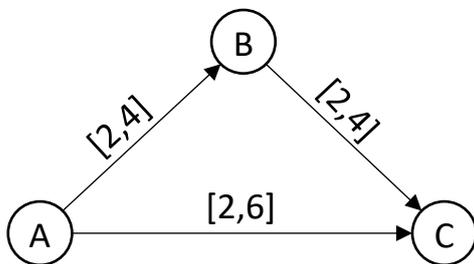
	A	B	C
A	0	4	6
B	-2	0	4
C	-2	-2	0

1.2 Consistency and minimal network

Temporal reasoning on STP is performed by propagating the constraints and obtaining the ***minimal network***. The minimal network is the tightest equivalent STP, i.e., an STP where the minimum and maximum implied distances between each pair of points are made explicit. Computing a minimal network of an STP corresponds to computing the *all-pairs shortest paths* of the graph; an algorithm such as the Floyd-Warshall's one can be used [15]. Such an algorithm can also determine the consistency of an STP by checking whether it contains *negative cycles* (i.e., cycles in the graph whose edges weights sum to a negative value). Floyd-Warshall's algorithm is shown below; in the algorithm 1, ... , n denote the time points (e.g., starting/ending points of actions) and D the matrix of distances.

```
Floyd-Warshall's algorithm
for k:=1 to n do
  for i:=1 to n do
    for j:=1 to n do
      D[i,j]=min(D[i,j],D[i,k]+D[k,j])
```

Property. Floyd-Warshall's algorithm is *correct* and *complete* on STP, i.e., it performs all and only the correct inferences while propagating the STP constraints [15]. Its temporal computational cost is *cubic* in the number of time points.

Applying Floyd-Warshall's algorithm to the STP *S* in the example allows to determinate the *minimal network* of *S*, where, for example, it is made explicit that, if B occurs at least 2 hours after A and C at least 2 hours after B, C must occur at least 4 hours after A.

	A	B	C
A	0	4	6
B	-2	0	4
C	-4	-2	0
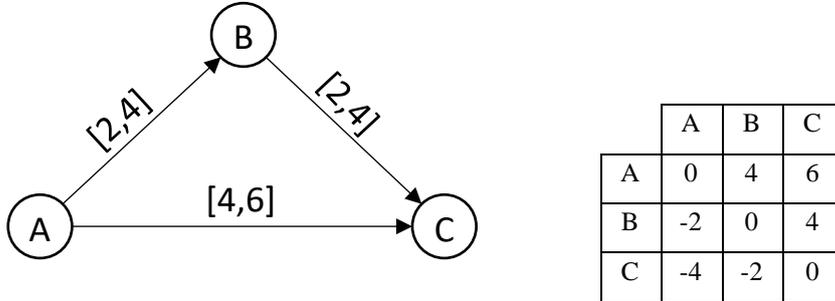
1.3 Solutions of an STP

Thanks to the properties of the minimal network, one is granted that each value of a constraint of the minimal network belongs to a solution of the STP. For example, given the constraint $2 \leq B{-}A \leq 4$ in the minimal network of the STP *S,* the value 3 for the distance between A and B (i.e., if B is exactly 3 hours after A), this value corresponds to the two solutions of the STP where (i) $C{-}B = 2$ and $C{-}A = 5$ and (ii) $C{-}B = 3$ and $C{-}A = 6$. However not every combination of values admitted by the constraints in the minimal network results in a solution of the STP. Consider, for example, $B{-}A = 3$ and $C{-}B = 4$; while they are individually admitted by the constraints in the minimal network of *S*, they cannot be extended to a solution: in fact, no consistent value can be chosen for $C{-}A$ (in fact $B{-}A = 3$ and $C{-}B = 4$ are values that belong to two different solutions).

If an STP changes because a new tighter constraint is added, a new constraint propagation is required because it is necessary to take into account the consequences of the change on the other constraints, which can possibly be tightened, and to reestablish the minimal network. For example, in *S,* if we tighten the constraint $B{-}A$ to $3 \leq B{-}A \leq 4$ (i.e., we rule out the value $B{-}A = 2$), also the other two constraints would be tightened and the new minimal network has $5 \leq C{-}A \leq 6$ (in fact, the value $C{-}A = 4$ is no longer possible) and $2 \leq C{-}B \leq 3$ (in fact, also the value $C{-}B = 4$ is no longer possible). On the other hand, a further tightening of the constraint between A and B such as $3 \leq B{-}A \leq 3$ would cause no further tightening of the other two constraints.

1.4 Query an STP

Given an STP, it is useful to ask queries and, in particular, whether some constraints are possible with regard to the STP (i.e., they are consistent with the STP or, equivalently, there is a solution of the STP where the constraints are satisfied) or are necessary with regard to the STP (i.e., the constraints are satisfied in every solution of the STP) [47]. In all cases, the below discussion assumes that the *minimal network* has already been obtained.

Queries about possibility with one constraint

Asking a "possible" query with one constraint is equivalent to determine whether the constraint is consistent with the minimal network of the STP, i.e., whether at least one value admitted by the query constraint belongs to a solution of the STP. In the example, asking whether a constraint such as $3 \leq B{-}A \leq 6$ is possible with regard to the STP *S*, implies to determine whether at least one value between 3 and 6 for $B{-}A$ belongs to a solution of the STP *S*. Thus, it is possible to answer to such a query by verifying whether the intersection between the query constraint and the corresponding constraint in the minimal network is empty. In the example $[3,6] \cap [2,4] \neq \emptyset$, thus the constraint is possible.

Queries about possibility considering more than one constraint

When the query is composed by more than one constraint, it is not possible to answer by simply inspecting the minimal network. In fact, for reasons derived from the discussions above, constraints can be individually consistent but inconsistent when considered together. For example, in *S* the

constraints $\{3 \leq B-A \leq 6, 4 \leq C-B \leq 4\}$ are individually possible but, taking them together, they do not correspond to any solution of S. Thus, in order to answer to "possible" queries with two or more constraints, such constraints must be added to the STP and then be propagated using the Floyd-Warshall's algorithm to detect whether a negative cycle has been created.

Queries about necessity with one constraint

The case where the query concerns whether one constraint is necessarily true in an STP corresponds to determine whether every solution of the STP satisfies it. Thus, it is sufficient to check whether the query constraint properly contains the corresponding constraint in the minimal network. For example, the query constraint $2 \leq B-A \leq 6$ is necessarily true in S, because $[2,6] \supseteq [2,4]$, thus the constraint is satisfied in each solution of S. On the other hand, the constraint $3 \leq B-A \leq 6$ is not necessarily true because there is at least a solution of S (i.e., the one where $B-A = 2$) where the query constraint is not satisfied.

Queries about necessity concerning more than one constraint

The case where the query concerns whether two or more constraints are necessarily true in an STP is asymmetric with regard to the "possible" query case since it can be answered without resorting to a further constraint propagation. In fact, to determine whether all the constraints are satisfied in each solution of an STP, it suffices to consider whether each constraint is necessarily true independently of the others [47]. For example, the constraints $\{2 \leq B-A \leq 6, 1 \leq C-B \leq 4\}$ are necessarily true in the STP S because both $2 \leq B-A \leq 6$ and $1 \leq C-B \leq 4$ are necessarily true (in fact, $[2,6] \supseteq [2,4]$ and $[1,4] \supseteq [2,4]$).

Appendix 2. Results of Temporal Reasoning (Minimal Networks)

In the following, for the sake of completeness, we report the minimal networks obtained in our experiments, when coping with the different examples in the paper.

	RT	W1$_P$	W2$_P$	WT$_S$	WT$_E$	AN2$_S$	AN2$_E$	DEC$_P$	ET$_S$	E1$_P$	ET$_E$	RM1$_S$	RM1$_E$
RT	0	**0**	1	**0**	∞	2	6	**0**	1	1 | 2	∞	2 | 3	8 | 9
W1$_P$	**0**	0	**1**	0	∞	2	6	0	1	1 | 2	∞	2 | 3	8 | 9
W2$_P$	-1	-1	0	-1	∞	**1**	**5**	-1	0	0 | 1	∞	1 | 2	7 | 8
WT$_S$	**0**	0	1	0	∞	2	6	0	1	1 | 2	∞	2 | 3	8 | 9
WT$_E$	-1	-1	0	-1	0	1	5	-1	0	0 | 1	∞	1 | 2	7 | 8
AN2$_S$	-1	-1	0	-1	∞	0	4	-1	0	0 | 1	∞	1 | 2	7 | 8
AN2$_E$	-2	-2	-1	-2	∞	-1	0	-2	-1	-1 | 0	∞	0 | 1	6 | 7
DEC$_P$	**0**	0	1	0	∞	2	6	0	1	1 | 2	∞	2 | 3	8 | 9
ET$_S$	0	0	1	0	∞	2	6	0	0	1	∞	2	8
E1$_P$	-1 | 0	-1 | 0	0 | 1	-1 | 0	∞	1 | 2	5 | 6	-1 | 0	0	0	∞	**1**	**7**
ET$_E$	-1 | 0	-1 | 0	0 | 1	-1 | 0	∞	1 | 2	5 | 6	-1 | 0	0	0	0	1	7
RM1$_S$	-1 | 0	-1 | 0	0 | 1	-1 | 0	∞	1 | 2	5 | 6	-1 | 0	0	0	∞	0	6
RM1$_E$	-3 | -2	-3 | -2	-2 | -1	-3 | -2	∞	-1 | 0	3 | 4	-3 | -2	-2	-2	∞	-2	0

Table A1. Minimal networks *all_mn* (first value in each cell) and *basic_mn* (second value in each cell) for Example 1 (query Q2). In case values are the same, they are reported just once. Underlined elements represent the LOG and ONTO constraints.

	RT	W1$_P$	W2$_P$	WT$_S$	WT$_E$	AN2$_S$	AN2$_E$	DEC$_P$	ET$_S$	E1$_P$	ET$_E$	RM1$_S$	RM1$_E$
RT	0	0	1	0	∞	2	6	0	1	2	∞	3	9
W1$_P$	0	0	1	0	∞	2	6	0	1	2	∞	3	9
W2$_P$	-1	-1	0	-1	∞	1	5	-1	0	1	∞	2	8
WT$_S$	0	0	1	0	∞	2	6	0	1	2	∞	3	9
WT$_E$	-1	-1	0	-1	0	1	5	-1	0	1	∞	2	8
AN2$_S$	-1	-1	0	-1	∞	0	4	-1	0	1	∞	2	8
AN2$_E$	-2	-2	-1	-2	∞	-1	0	-2	-1	0	∞	1	7
DEC$_P$	0	0	1	0	∞	2	6	0	1	2	∞	3	9
ET$_S$	0	0	1	0	∞	2	6	0	0	1	∞	2	8
E1$_P$	0	0	1	0	∞	2	6	0	0	0	∞	1	7
ET$_E$	0	0	1	0	∞	2	6	0	0	0	0	1	7
RM1$_S$	0	0	1	0	∞	2	6	0	0	0	∞	0	6
RM1$_E$	-2	-2	-1	-2	∞	0	4	-2	-2	-2	∞	-2	0

Table A2. Minimal network calculated to answer query Q1.

	RT	CC$_P$	UA$_S$	UA$_E$	NA$_P$	NAA$_S$	NAA$_E$
RT	0	14	15	20	∞	∞	∞
CC$_P$	-12	0	1	6	∞	∞	∞
UA$_S$	-12	0	0	5	∞	∞	∞
UA$_E$	-16	-4	-4	0	∞	∞	∞
NA$_P$	∞	∞	∞	∞	0	1	5
NAA$_S$	∞	∞	∞	∞	-1	0	4
NAA$_E$	∞	∞	∞	∞	-5	-4	0

Table A3. Minimal network obtained from the extended temporal reasoning for Example 2. CC_P and NA_P: are the times of execution of the punctual actions Calcium Carbonate and Nalidixic Acid administrations; UA_S and UA_E are the endpoints of the urine alkalinization effect of the Calcium Carbonate; NAA_S and NAA_E: are the endpoints of the Nalidixic Acid absorption.

	RT	CC$_P$	UA$_S$	UA$_E$	NA$_P$	NAA$_S$	NAA$_E$
RT	0	14	15	20	∞	∞	∞
CC$_P$	-12	0	1	6	∞	∞	∞
UA$_S$	-12	0	0	5	∞	∞	∞
UA$_E$	-16	-4	-4	0	∞	∞	∞
NA$_P$	-18	-6	-5	0	0	1	5
NAA$_S$	-19	-7	-6	-1	-1	0	4
NAA$_E$	-23	-11	-10	-5	-5	-4	0

Table A4. Minimal network obtained by adding the first element of the disjunction in Condition (4) to the minimal network in Table A3.

	RT	CC$_P$	UA$_S$	UA$_E$	NA$_P$	NAA$_S$	NAA$_E$
RT	0	14	15	20	21	22	26
CC$_P$	-14	0	1	6	7	8	12
UA$_S$	-14	0	0	5	7	8	12
UA$_E$	-18	-4	-4	0	3	4	8
NA$_P$	-20	-6	-5	0	0	1	5
NAA$_S$	-21	-7	-6	-1	-1	0	4
NAA$_E$	-25	-11	-10	-5	-5	-4	0

Table A5. Minimal network of Example 2 after the propagation of the refinement constraints.