

Piovesan, Luca; Molino, Gianpaolo; Terenziani, Paolo

Article

An ontological knowledge and multiple abstraction level decision support system in healthcare

Decision Analytics

Provided in Cooperation with:

Springer Nature

Suggested Citation: Piovesan, Luca; Molino, Gianpaolo; Terenziani, Paolo (2014) : An ontological knowledge and multiple abstraction level decision support system in healthcare, Decision Analytics, ISSN 2193-8636, Springer, Heidelberg, Vol. 1, Iss. 1, pp. 1-24, <https://doi.org/10.1186/2193-8636-1-8>

This Version is available at:

<https://hdl.handle.net/10419/161815>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by/4.0/>

RESEARCH

Open Access

An ontological knowledge and multiple abstraction level decision support system in healthcare

Luca Piovesan^{1*}, Gianpaolo Molino² and Paolo Terenziani³

Abstract

The rationalization of the healthcare processes and organizations is a task of fundamental importance to grant both the quality and the standardization of healthcare services, and the minimization of costs. Clinical Practice Guidelines (CPGs) are one of the major tools that have been introduced to achieve such a challenging task. CPGs are widely used to provide decision support to physicians, supplying them with evidence-based predictive and prescriptive information about patients' status and treatments, but usually on individual pathologies. This sets up the urgent need for developing decision support methodologies to assist physicians and healthcare managers in the detection of interactions between guidelines, to help them to devise appropriate patterns of treatment for comorbid patients (i.e., patients affected by multiple diseases).

We identify different levels of abstractions in the analysis of interactions, based on both the hierarchical organization of clinical guidelines (in which composite actions are refined into their components) and the hierarchy of drug categories. We then propose a general methodology (data/knowledge structures and reasoning algorithms operating on them) supporting user-driven and flexible interaction detection over the multiple levels of abstraction. Finally, we discuss the impact of the adoption of computerized clinical guidelines in general, and of our methodology in particular, for patients (quality of the received healthcare services), physicians (decision support and quality of provided services), and healthcare managers and organizations (quality and optimization of provided services).

Keywords: Computer-interpretable guidelines; Healthcare decision support; Ontology of interactions; Interaction detection algorithm; Multiple level analysis

Background

Given its social and economic relevance, the healthcare system is the object of continuous studies, aiming at optimizing it, both in terms of costs, and of the quality of the supplied services. ICT can significantly contribute to achieve healthcare optimization, both in terms of providing healthcare managers and organizations with decision analytic tools to optimize healthcare processes in terms of process and cost rationalization; and also in terms of providing clinical decision-making support, improving the quality and the standardization of healthcare services. Clinical Practice Guidelines (CPGs) are

one of the major tools that have been introduced to provide medical decision support. CPGs are, in the definition of the USA Institute of Medicine, "*systematically developed statements to assist practitioner and patient decisions about appropriate health care in specific clinical circumstances*" (Institute of Medicine, Committee on Quality Health Care in America 2001). Thousands of CPGs have been devised in the last few years. For instance, the Guideline International Network (<http://www.g-i-n.net>) groups 77 organizations from 4 continents, and provides a library of more than 5000 CPGs. CPGs are commonly recognized as a tool to encode and support the practical adoption of evidence-based medicine (EBM).

The adoption of computerized approaches to acquire, represent, execute and reason with CPGs can further

* Correspondence: piovesan@di.unito.it

¹Department of Computer Science, Università degli Studi di Torino, C.so Svizzera, 185, Turin 10135, Italy

Full list of author information is available at the end of the article

increase the advantages of CPGs, providing crucial advantages to:

1. patients, enabling them to receive the best quality medical treatments (since CPGs are actually a way of putting EBM into practice);
2. physicians, providing them with a standard reference which they may consult, with a way of certifying the quality of their activity (e.g., for insurance or legal purposes), as well as with advanced support for their decision-making activity;
3. hospitals, and healthcare centers, providing them with tools to enable the quality and the standardization of their services, as well as with a means of evaluating quality, and of optimizing costs and resources.

In recent years, the research about computerized guidelines has reached an important role within the Medical Informatics community, and many different approaches and projects have been developed to create domain-independent computer-assisted tools for managing, acquiring, representing and executing computer-interpretable clinical guidelines (henceforth CIGs). See e.g. the systems Asbru (Shahar et al. 1998), EON (Shahar et al. 1996), GEM (Shiffman et al. 2000) GLARE (Terenziani et al. 2001) (Terenziani et al. 2003), GLIF (Peleg et al. 2000), GUIDE (Quaglini et al. 2001), PROforma (Fox et al. 1998), and the collections (Gordon and Christensen 1995) (Fridsma 2001) (Ten Teije et al. 2008) (Peleg 2013). One of such approaches is GLARE (Guideline Acquisition, Representation and Execution), which started from 1997 in a long-term cooperation between the Department of Computer Science of the University of Eastern Piedmont Alessandria, Italy, and the *Azienda Ospedaliera San Giovanni Battista* in Turin (one of the largest hospitals in Italy). Besides supporting CIG acquisition, representation, storage and execution, GLARE is characterized by the adoption of advanced Artificial Intelligence and Temporal Database formal techniques to provide advanced supports for different tasks, including reasoning about temporal constraints (Anselma et al. 2006), the treatment of periodic data (Stantic et al. 2012), guideline versioning (Anselma et al. 2013), model-checking verification (Bottrighi et al. 2010), decision support (based on Decision Theory) (Montani and Terenziani 2006) (Anselma et al. 2011), contextualization (Terenziani et al. 2004).

A critical issue: the treatment of comorbid patients

By definition, clinical guidelines address specific clinical circumstances (i.e., specific diseases). However, unfortunately, specific patients may be affected by more than one disease. The treatment of patients affected by multiple diseases (*comorbid* patients) is one of the main challenges

for modern healthcare, also due to population aging and the increase of chronic diseases. For example, the prevalence of chronic diseases such as hypertension, diabetes mellitus or heart failure in society has grown during the last decades as the number of elderly people has increased (Robert Wood Johnson Foundation and Partnership for Solutions 2004). Moreover, most of the patients suffering from such chronic diseases, are affected by two or more of them (Gijsen et al. 2001).

Thus, “health care systems must deal with an increasing number of patients with several simultaneous pathologies (i.e., co-morbid 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”. (Riaño and Collado 2013).

However, though some CPGs covering frequently occurring co-morbidities 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 co-morbid 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”. (Michalowski et al. 2013).

Several studies show that the lack of such methods is one of the obstacles to the adoption of CPGs in clinical practice, and the development of these methods has been identified as one of the “grand challenges” for clinical decision support (Sittig et al. 2008).

In the past few years, some computer-based approaches have started to face this problem, aiming at providing physicians with different forms of support for managing multiple CIGs. Notably, in the last Artificial Intelligence in Medicine Europe (AIME’13) Conference, held in Murcia, Spain, from 29/5/2013 to 1/6/2013, a whole session (4 papers) was devoted to such an issue (Sánchez-Garzón et al. 2013) (Jafarpour and Abidi 2013) (Riaño and Collado 2013) (Michalowski et al. 2013). The state of the art in the computer-based treatment of co-morbidities is presented in section “Related Works”. Despite the high quality of the recent research approaches in the literature, the treatment of co-morbidities is an extremely difficult and challenging problem, so that substantial steps forward need to be achieved. In the rest of the paper, we present our contribution.

Goals of our overall approach

The long-term goal of our overall approach is that of extending GLARE providing physicians with a domain-independent and guideline-independent set of tools and methodologies to facilitate the integration of two or more CIGs, by supporting in (i) the detection of the interactions between two or more guidelines, (ii) the solution of the interactions, and (iii) the merging of multiple guidelines in the treatment of the specific patient at hand.

Our overall goal is that of providing a suite of tools and methodologies to support physicians, providing them, as much as possible, with information and hints that may be helpful in their activity when facing multiple guidelines. As a consequence, there are several *desiderata* that our approach will need to meet:

Interactivity: we think that the system should help the physician in the merging process, but it shouldn't replace her/him, so the system must involve the user in decisions and must be easy to use;

Flexibility: we do not think there can be a unique "golden" standard methodology to merge two guidelines. Indeed, we aim at supporting physicians by providing them with physician-driven explorations of the different possible ways of facing the problem;

Multiple abstractions: given the complexity of the overall task, physicians usually face it in a "top-down" fashion, operating at multiple levels of abstractions; supporting user-driven interactive and flexible merging of CIGs operating at different detail levels (along different dimensions) is one of the core goals of our approach.

In this paper, we will focus only on the first, essential, step in the development of the above approach, namely the development of a methodology addressing the *detection of the interactions* between different CIGs. It is worth stressing that there are already two kinds of tools that physicians use in order to supervise treatments and detect drug interactions: information tools like PDAs on online access to pharmacological databases (Robert Wood Johnson Foundation and Partnership for Solutions 2004), and alerting systems such as CPOE (Koppel et al. 2005), that detect and inform about interactions (Medscape (<http://reference.medscape.com/drug-interactionchecker/>), Drugs.com (http://www.drugs.com/drug_interactions.html), etc.). However, such tools mostly focus on drug-drug interactions only. Unfortunately, this is only a very limited support, when a physician has to detect and analyze the interactions between two or more guidelines. Indeed, though a large variety of representation formalisms exists (Ten Teije et al. 2008), most CIG formalism support a hierarchical decomposition of guidelines at multiple levels of detail, in which composite actions may be represented, and then refined (possibly at different levels of abstraction) into their components. At the finest level of detail therapeutic actions in the guideline may recommend,

depending on the accuracy, the use of drugs or drugs categories, or active principles (thus, also the interactions between drug categories must be considered). However, at higher levels, composite actions may be better characterized on the basis of their intentions (i.e., of the high-level goals they aim to achieve; consider, e.g., ASBRU (Shahar et al. 1998)). We aim at devising a more suitable and complete support to the detection of interactions between guidelines, in which also high-level action intentions are taken into account, and the interaction analysis is supported at any possible level of abstraction considered in the guideline specification. As previously discussed, besides multiple abstractions, interactivity and flexibility are essential *desiderata* of our approach. Given the relevance of such *desiderata*, we illustrate the need for them through an example.

An abstract example of flexible and interactive interaction detection and analysis

The example here is deliberately abstract, to stress the generality of our methodology (real medical examples will be widely provided throughout all the rest of the paper). In the example, as well as in the approach described in this paper, we focus our attention on the actions in the therapeutic parts of CIGs (composite actions and pharmacological actions, recommending drug administrations).

Example: Suppose, for instance, that the user physician UP needs to detect the interaction between two composite actions A1 (belonging to the guideline G1) and A2 (belonging to the CIG G2). (S)he selects A1 from G1, and A2 from G2, and then looks for the interactions. At this level of abstraction, i.e., the interaction between the intentions of A1 (*int1* in the upper part of Figure 1) and A2 (*int2* in Figure 1) is taken into account.

The possible continuation of the interaction detection process may depend strictly on the output of such an analysis. For instance:

- Case 1 (shown in Figure 1) Suppose that, at this level of abstraction, no interaction is found between *int1* and *int2*. UP wants to further refine her search. (S)he expands A1 and A2 into the actions composing them in the CIGs, focuses on some of them (see the lower part of Figure 1), e.g., A₁₁ and A₁₂ (one of the possible decompositions of A1 in G1, or part of it) and A₂₁ and A₂₂ (one of the possible decompositions of A2 in G2, or part of it), which are all pharmacological actions. Then, UP looks for the interactions between them. In particular, pharmacological actions prescribe the administration of drug categories (for example, in Figure 1, we represent the fact

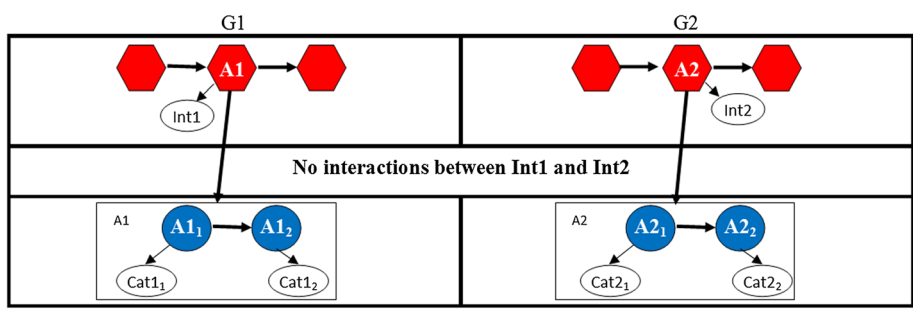


Figure 1 States of guideline explorations for Case 1.

that $A1_1$ prescribes the administration of $cat1_1$, and so on), and the interactions between them is further analyzed.

Case 1.1 (shown in Figure 2) Suppose that no interaction between the drug categories ($cat1_1$ and $cat1_2$ from G1, $cat2_1$ and $cat2_2$ from G2) is found. UP further wants to refine his search by expanding the pharmacological actions to the level of drugs, selecting few drugs (e.g., $Drug_b$ for $Cat1_1$, etc., in the graphical example in Figure 2), and, finally, analyzing their interactions. This process can *iteratively go on* until non-interacting drugs are found for the four actions.

Case 1.2 (shown in Figure 3) On the other hand, suppose that some interaction between the drug categories ($cat1_1$ and $cat1_2$ from G1, $cat2_1$ and $cat2_2$ from G2) is found. UP may use the focusing tool to “go up” to A1 and A2, and focus on an alternative decomposition of them (e.g., in Figure 3, $A1_3$, $A1_4$ and $A1_5$ for A1, $A2_3$ and $A2_4$ for A2). The process continues with the analysis of the interactions between $A1_3$, $A1_4$ and $A1_5$ (from G1) and $A2_3$ and $A2_4$ (from G2).

Notice that, although the above example seems to indicate that interactions are always “bad” (so that they are

a reason for looking for alternatives), this is not necessarily true. As we will analyze in Section “Interactions Ontology”, the spectrum of possible interactions types is wide, and only a few types can be easily categorized as “bad” (e.g., two intentions are “opposite”), or “good” (e.g., two intentions are “equal”, or are “independent”). In all the other cases, interactions are “workable” (e.g., a “reinforcing” interaction between two drugs can be managed by operating on the posology of such drugs), and the user-physician may choose to: (i) manage and “solve” them, e.g., through the modification of drug posology, (ii) try to avoid them, looking for alternatives, or (iii) further investigate them (e.g., at a lower level of detail).

The above example clearly shows that the detection of interaction is intrinsically an *interactive* process, in which each step is “unpredictable”, since it depends on the output of the previous step. Additionally, the above consideration shows, that, given the output of one step, there are (in most cases) alternative ways of going on, and it is almost impossible to automatically choose between them (indeed, user-physicians might accept knowledge-based suggestions as the way to go on, but certainly not a system-made decision).

Thus and *interactive* and *user-driven* interaction detection process are required. The goal of the work we present in this paper is to identify a general methodology (data/knowledge structures and reasoning algorithms operating on them) supporting it.

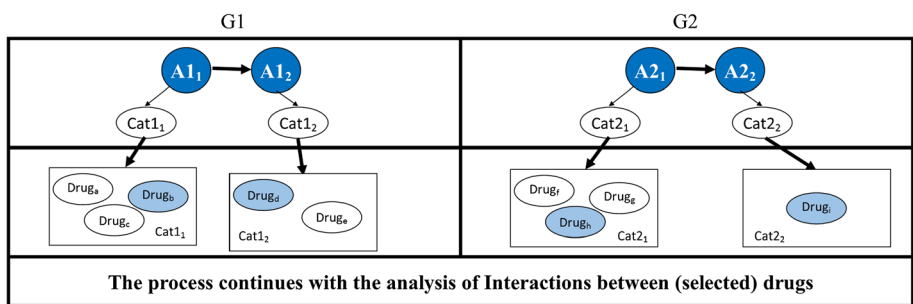


Figure 2 States of guideline explorations for Case 1.1.

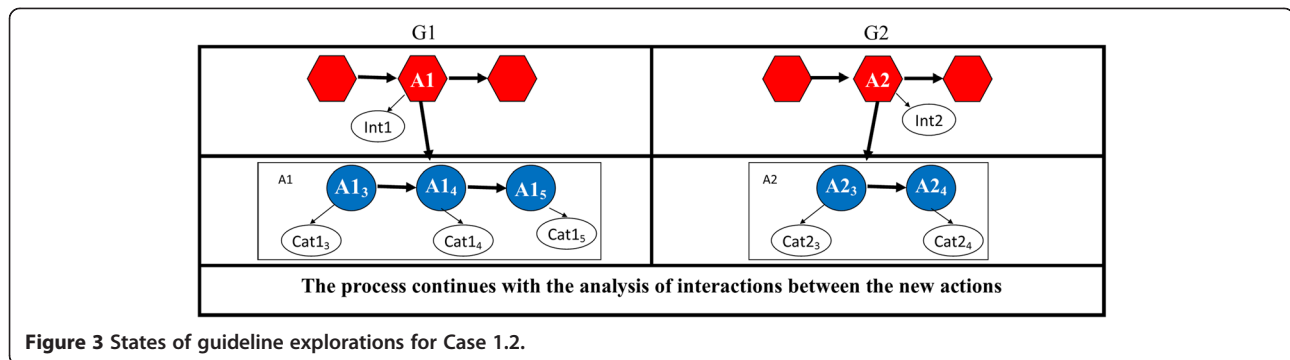


Figure 3 States of guideline explorations for Case 1.2.

A digression: knowledge-based decision support

Before starting the technical discussion, it is worth opening a parenthesis to clarify the relationships between the decision-support methodology we propose in this paper and “standard” decision analytics methodologies. In this paper, we neither directly apply statistical methods, nor do we propose cost/risk – utility/benefit analysis techniques. Indeed, the input of our approach is not a (big) set of *data*, but two (or more) CPGs, and a medical ontology, i.e., three (or more) different sets of *knowledge*. Since guidelines and ontologies are usually quite large bodies of knowledge, a “flat” analysis of it would be difficult, if not totally unfeasible, to users. We thus propose a new decision support methodology, based on multiple levels of abstraction on medical knowledge, to deal with co-morbid patients. In particular, we focus on developing software tools and methodologies to support the analysis of interactions between CPGs, considering also ontological knowledge. Since CPGs and ontologies are pieces of knowledge, our methodology is necessarily *knowledge-based* (as opposed to statistical-based methodologies). Nevertheless, it is important to remember that statistical methods play a fundamental role in the CPG context. As a matter of fact, CPGs are grounded on the principles of *evidence-based* medicine, which dictates that statistical and decision analytics methods must be applied on clinical trials as a necessary preliminary step to the identification and representation of the evidence-based recommendations they contain. Indeed, CPGs could thus be seen, informally speaking, as the symbolic representation of best-practice patterns of clinical processes built on the basis of the outcome of statistical analytics on clinical trials.

Thus, CPGs in general, and our approach in particular, are used to provide *knowledge-based decision support, grounded on statistical evidence*.

As a further difference with respect to “traditional” decision analytics methods, the main intended users of CPGs are physicians. However, in Section “Impact on physicians, patients and managers”, we will discuss in more detail the possible impacts of the adoption of our

approach on physicians, healthcare managers, and patients, highlighting how healthcare managers can take advantage of it.

Summary

The paper will be organized as follows. The section “The GLARE formalism” presents the GLARE approach, with specific emphasis on the physician-friendly formalism we have devised in order to represent CIGs. The first step of our approach is the definition of appropriate formalisms to cope with the knowledge that is needed to support the analysis of interactions (Section “Interactions: ontological notions”). In Section “Interactions detection algorithm”, we describe a new methodology to explore the above-mentioned knowledge sources, and multiple guidelines, in an interactive and flexible way. In Section “Impact on physicians, patients and managers”, we discuss the impact of CIGs, and of our approach, in different healthcare contexts. In Section “Related works”, we will describe the state-of-the-art, and, finally, we draw the conclusions.

Methods

Main features of the GLARE system

GLARE is a long-term project started in 1997 to provide a user-friendly and domain-independent tool to support CIG acquisition, representation and execution, and to provide the physician with a wide range of facilities, helping her/him in decision support or CIG analysis tasks. The main goals of the GLARE systems are:

- to be **domain-independent**. In particular, GLARE has been already applied to deal with a wide range of CIG, ranging, e.g., from asthma to ischemic stroke.
- to be **user-friendly**. GLARE is intended to be a tool to support physicians (in particular, regarding decision making), not at all to replace them. Such a goal has several implications, including the facts that (i) CIGs representation language must be as close as

possible to the physicians' way of looking at it, and (ii) the "technological" complexity of the system must be hidden to the user-physicians, through a user-friendly and easy-to-use interface.

- to be easily **extendable**. Indeed, physicians should be provided with plenty of facilities concerning CIG treatment. Thus, the system architecture must be modular and easily extendable.

Of course, each one of the above tasks has deeply influenced the design of GLARE, and its main features.

GLARE representation language

In the medical literature, clinical guidelines are often described using different levels of detail (and top-down refinement), and oriented graphs of actions, similar to flow-charts are often used to describe the ordering of actions at the different levels. As a consequence, 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.

In GLARE, we distinguish 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.

In order to be as physician-friendly as possible, a limited and "physician oriented" number of types of atomic actions have been identified:

- *work actions*, i.e. actions that describe a procedure which must be executed at a given point of the GL;
- *pharmaceutical actions*, specifying a drug (or drug category, of active principle) to be administered to the patient, and the posology.
- *decision actions*, used to model the selection among different alternatives. We have introduced a distinction between:
 - *diagnostic decisions*, used to make explicit the identification of the disease the patient is suffering from, among a set of possible diseases, compatible with her findings. A diagnostic decision is represented as an open set of triples <diagnosis, parameter, score> (where, in turn, a parameter is a triple <data, attribute, value>), plus a threshold to be compared with the different diagnoses' scores;
 - *therapeutic decisions*, used to represent the choice between paths in a GL, where each path represents a particular therapeutic process. The choice can be made by evaluating a fixed set of parameters (effectiveness, cost, side-effects, compliance and duration);

- *query actions*, i.e. requests of information (typically patient's parameters), that can be obtained from the outside world (physicians, databases, patient visits or interviews). The GL execution cannot go on until this information has been obtained;
- *conclusions*, which explicitly identify the output of a decision action.

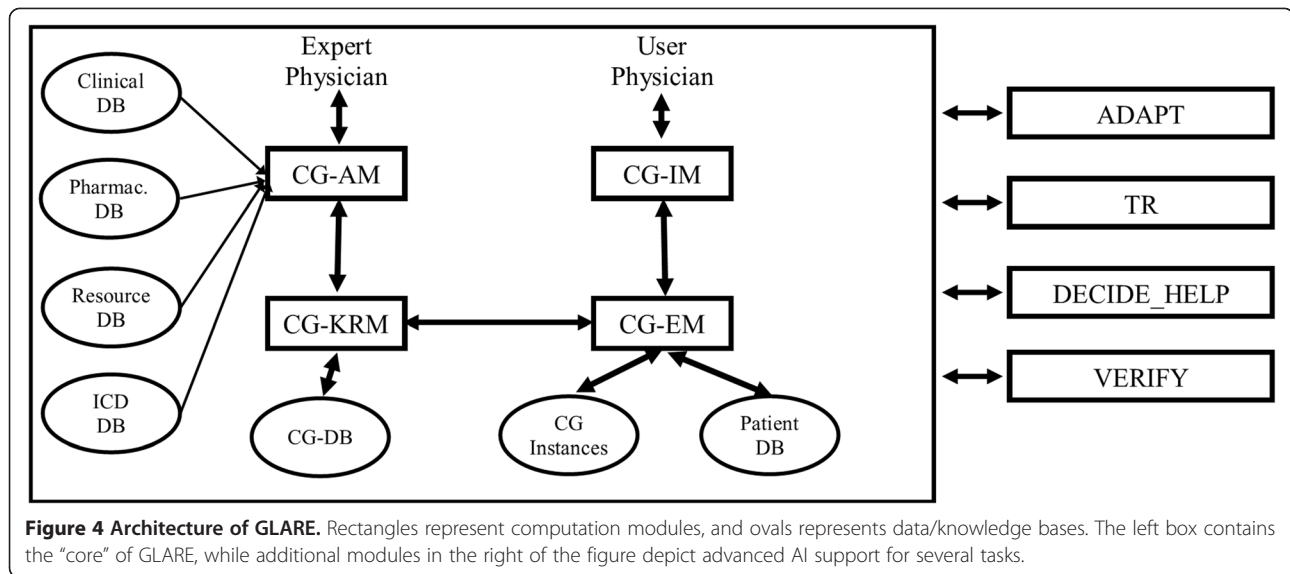
In particular, the distinction between diagnostic and therapeutic decisions is very fundamental to physicians, and constitutes one of the features that distinguish GLARE from most CIG approaches in the literature. Each kind of action is described by a set of properties. For instance, a pharmaceutical action is described by:

- *Action Type*, which for each action of this category takes the value "Administration of Medicament",
- *Substance*, which describes the administered drug,
- *Posology*, which describes the dosage and the modality of assumption. This property structure is complex, since it might involve the description of periodicity (Anselma et al. 2006).

Actions in a GL 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 is the following action to be executed; 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. In detail, given a repeated action, the number of its repetitions can be fixed a priori, or, alternatively, it can be asserted that the action must be repeated until a certain *exit condition* becomes true (in this case, the number of repetitions is only known at runtime, during execution). In particular, advanced AI techniques are required to deal with temporal reasoning in repeated actions (Anselma et al. 2006).

Architecture of the kernel of GLARE

The core of GLARE (see box on the left of Figure 4) is based on a modular architecture. CG_KRM (Clinical Guidelines Knowledge Representation Manager) is the main module of the system: it manages the internal representation of CPG, and operates as a domain-independent and task-independent knowledge server for the other modules; moreover, it permanently stores the acquired CPG in a dedicated Clinical Guidelines Database (CG-DB). The Clinical Guidelines Acquisition Manager (CG_AM) provides expert-physicians with a user-friendly graphical interface to introduce the CPG into the CG_KRM and describe them. It may interact with four databases: the Pharmacological DB, storing a structured list of drugs and their costs; the Resources DB, listing



the resources that are available in a given hospital (it is therefore used to represent the context-dependent version of a CPG); the ICD DB, containing an international coding system of diseases; the Clinical DB, providing a “standard” terminology to be used when building a new CPG, and storing the descriptions and the set of possible values of clinical findings.

The execution module (CG-EM) executes a CPG for a specific patient, considering the patient’s data (retrieved from the Patient DB). The schema of the Patient DB mirrors the schema of the Clinical DB. Therefore, the inter-action with the Clinical DB during the acquisition phase makes it possible to automatically retrieve data from the Patient DB at execution time. CG-EM stores the execution status in another DB (CG Instances) and interacts with the user-physician via a graphical interface (CG-IM).

It is worth noticing that, through the CG-AM and CG-IM modules, GLARE provides user-friendly interfaces to user, achieving the goal of “hiding” them in the internal complexity of the system.

Extended architecture

To enhance extendibility, GLARE’s architecture is open: new modules and functionalities can be easily added to the system if/when necessary. In recent years, we have added new modules and/or methodologies in order to cope with automatic resource-based contextualization (ADAPT module, (Terenziani et al. 2004)), temporal reasoning (TR, (Anselma et al. 2006) and (Stantic et al. 2012)), decision-making support (DECIDE_HELP, (Montani and Terenziani 2006) and (Anselma et al. 2011)), and model-based verification (VERIFY, (Bottrighi et al. 2010)). While all the methodologies to cope with the above issues have been widely addressed, not all the prototypical software modules

have been implemented yet. However, we envisage a modular architecture, in which all such modules will be loosely coupled with the core of GLARE, as shown in the right part of Figure 4.

Since the focus of this paper is decision support, in the following we briefly explain our decision-making facility (see (Montani and Terenziani 2006) and Anselma et al. (2011) for more details).

Decision-theory-based decision support in GLARE

When executing a CIG on a given patient, a physician can be faced with a choice among different therapeutic alternatives, and identifying the most suitable one is often not straightforward. Actually in several situations no alternative is really “better” than the others, from a strictly clinical viewpoint, and CIGs are only meant to present all the range of choices, leaving the user the responsibility of selecting the “right” one.

In clinical practice, various selection parameters (such as the costs and effectiveness of the different procedures) are sometimes available when executing a CIG, but the task of comparing and balancing them is typically left to the user. Decision theory seems a natural candidate as a methodology for supporting this analysis; to this hand, we have created a mapping between the CIG representation primitives and decision theory concepts, and a decision theory tool for supporting therapy selection in GLARE.

In short, in a well-formed CIG, a therapeutic decision action is always preceded by a query action, which is adopted to collect all the patient’s parameters necessary (and sufficient) for taking the decision itself. Each therapeutic decision is therefore based on an (explicit or implicit) data collection completed at decision time, and does not depend on the previous history of the patient.

We can thus say that the CIG describes a first-order Markov model, since each time a query action is implemented, the patient's situation is completely re-assessed, and an (explicit or implicit) query action is always found before a decision action. This observation allows us to represent a CIG as a Markov Decision Process (MDP), which has been recognized as a basic representation framework for dynamic decision making under uncertainty.

In particular, it is straightforward to define the concept of state as the set of patient's parameters that are normally measured for taking decisions and for assessing therapy outcomes. On the other hand, we consider work actions as the means of producing state transitions, since they are the only actions with a potential effect on the state variables. The utility of reaching a state can be evaluated in terms of life expectancy, corrected by Quality Adjusted Life Year (QALYs) (Gold et al. 1996). We can derive the utility of a state from the medical literature, as we do for obtaining the probability of state transitions. Costs can be interpreted as monetary expenses: each work action will have a price. Additionally, costs can also be evaluated in terms of the time and the resources required to complete work actions.

On the basis of these preliminary mapping considerations, which are general enough to be easily applied within any of the systems for the computerized management of CPGs described in the literature, we are implementing a decision theory facility in GLARE. In our approach, the MDP describing the CIG is represented resorting to a dynamic decision network (Tatman and Shachter 1990), a choice that allows one to explicitly take advantage of conditional independencies from the modeling viewpoint. In particular, the GLARE decision theory facility will enable the user: (1) to identify the optimal policy, and (2) to calculate the expected utility along a path, by exploiting classical powerful dynamic programming algorithms.

Interactions: ontological notions

Motivations for and advantages of an ontology-based approach

The first step of our approach is the definition of an ontology, clarifying the semantics of interactions, and modeling the different knowledge sources involved in their detection and analysis. In health informatics, ontologies are defined as “collections of formal, machine-processable and human interpretable representation of the entities, and the relations among those entities, within a definition of the application domain” (Rubin et al. 2006).

Since 2001, there has been an increasing use of ontological approaches to health (Liaw et al. 2013). As a matter of fact, the adoption of formal ontologies is almost becoming standard in medical informatics, since it provides many advantages, including terminological standardization and

the definition of a common semantics. For instance, SNOMED (<http://www.ihtsdo.org/snomed-ct/>) is a widely used clinical terminology, and stresses that “the use of a standard clinical terminology makes a significant contribution towards improving the quality and safety of health-care by enabling consistent representation of meaning in an electronic health record”. In particular, the definition of an ontology for interactions allows us to achieve several different goals:

Define shared semantics for guidelines and interactions: we need to define a semantic model for interactions and guidelines actions. Such semantic model must be shared and “understood” both by information systems and by human users (Gruber 2009; Rubin et al. 2006).

Define a standard for terminology and representation: GLARE is a standard for guidelines representation. The use of ontologies for the action model supports the standardization between different guidelines. In addition, ontologies are often used to disambiguate terms. The use of existing and consensus ontologies for data representation avoid ambiguities and allows the reuse of the knowledge.

Unify different sources: in the medical field, data can derive from different sources. Unify them using ontologies helps to improve data quality (Liaw et al. 2013).

Reason on the data: the use of standard ontology representation as RDF or OWL allows processing the content of information through existing reasoners (McGuinness and van Harmelen 2004): “by incorporating defined rules, ontologies may also generate logical inferences” (Pérez-Rey et al. 2006).

Interactions and levels of abstraction

Our ontology of interactions is based on the consideration that physicians usually analyze the interactions between guidelines at different levels of detail. At the highest level, they take into account the intentions (or goals) of actions (see, e.g., examples 1 and 2 in the following). Then, moving towards more specific information, they may also consider the interactions between specific drugs categories (e.g., Example 3), and, finally, between specific drugs (e.g., Example 4).

Example 1. Intentions Interaction: in the treatment of edema due to heart failure, the main intention is to decrease the accumulation of fluids. In the case in which that particular patient also manifests a syncope crisis, it is likewise necessary to increase the blood pressure. However, this later intention forces the heart to a higher activity, which in turn may increase the heart failure, worsening the edema.

Example 2. Intentions Interaction: in patients affected by thrombosis, anticoagulant drugs are usually administered, in order to decrease the formation of thrombi. If the patient affected by thrombosis has also a risk of

bleeding (e.g. due to a minor surgery) the anticoagulant therapy must be discontinued because of the need to increase the blood coagulation.

Example 3. Drug Categories Interaction: the administration of antacids (category) at the same time of other drugs orally administered can change the intestinal absorption of the later decreasing their effects (e.g. levofloxacin).

Example 4. Drugs Interaction: in many cases, interactions involve individual drugs. For example, between trimethoprim and sulfamethoxazole there is a strengthening synergism: the antibiotic effect of their concomitant administration is stronger than the sum of the two single effects.

Our goal is to devise a general ontology underlying the detection of guideline interactions. Our methodology is completely general, and we are not committed to any specific pre-existent ontology. However, given its relevance and generality, we use the SNOMED ontology (<http://www.ihtsdo.org/snomed-ct/>) as a reference one. Sometimes we also consider the Anatomical Therapeutic Chemical (ATC) ontology (<http://www.whocc.no/atc>) as concerns the drug classification. We extend them in order to model both the different sources of interactions (intentions, drugs categories, drugs) and the different types of interactions between them.

For the sake of interaction detection, the description of guideline actions must include the intention of composite and work actions, or the drug (class of drugs, active principle) of pharmacological actions. We first describe our ontology of intentions (subsection “Intentions Ontology”) and of drug classes (subsection “Drug Classes Ontology”). Then, in subsection “Interactions Ontology”, we describe our ontology for the interactions (between intentions, and between drug classes and drugs).

Intentions ontology

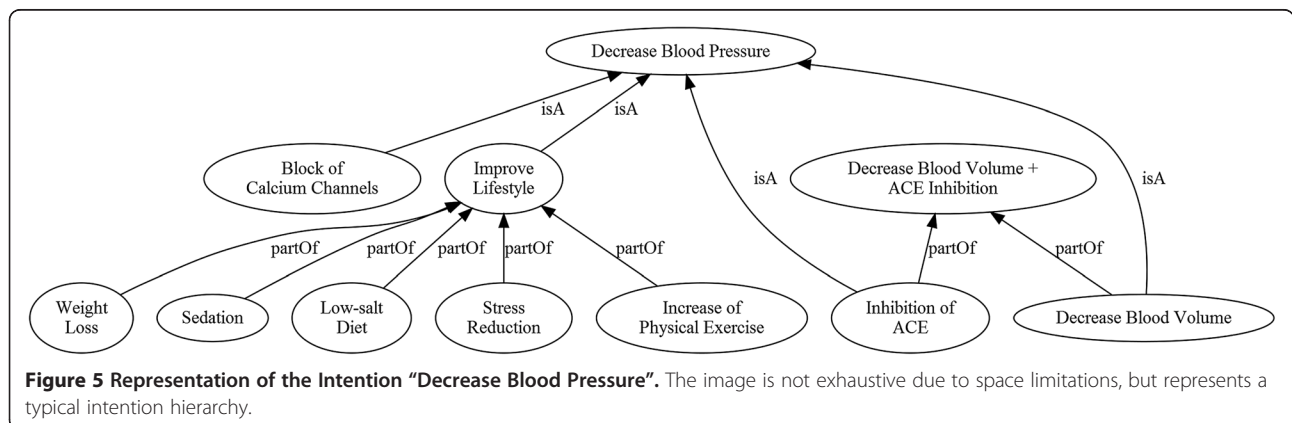
At a high level of abstraction, an action can be represented by its objectives or (desired) effects. In this section, we focus on the objectives of composite actions (plans), work actions and pharmacological ones, and we

call them *Intentions* (i.e., the goals that the physician expects to obtain realizing that action)^a.

Intentions can be organized along a hierarchy of ISA and PART-OF relations: high-level intentions can be decomposed into lower-level intentions, and alternative decompositions are possible.

For instance, Figure 5 shows (not exhaustively) the hierarchy of the intention “Decrease Blood Pressure”. That intention is typically obtained through the realization of several lower-level intentions and many alternative decompositions (and combinations between them) are possible. Then, lower level intentions can be decomposed by many part-of relations (as can be seen for the “Improve Lifestyle” intention), or not (for example “Block of Calcium Channels”). A particular case is represented by the combination of more intentions in order to reinforce the higher-level one: the intention “Inhibition of Angiotensin Converting Enzyme (ACE)” is both a sub-intention of “Decrease Blood Pressure” and a part of the combined intention “Decrease Blood Volume + ACE Inhibition”.

Ultimately, at the lower level, the intentions can be described as (desired) modifications of specific parameters of the patient status. In the following, we call such intentions *atomic intentions*. In our ontology, atomic intentions can be represented by a couple <BodyPartAttribute, Modality> in which BodyPartAttribute is the attribute that is expected to change, and Modality is an element belonging to the set {Increasing, Decreasing, Stability} and describes how the attribute is expected to change. Of course, if needed, the hierarchy of modalities might be further refined. For instance, we can distinguish between different degrees of increase/decrease. Figure 6 shows that BodyPartAttribute is in its turn connected with the respective body part. Both concepts have a counterpart in SNOMED Ontology: BodyPart is equivalent to the SNOMED concept “Body Structure” and BodyPartAttribute to the SNOMED concept “Observable Entity”, whilst concepts of Intention and Modality are not included in SNOMED.



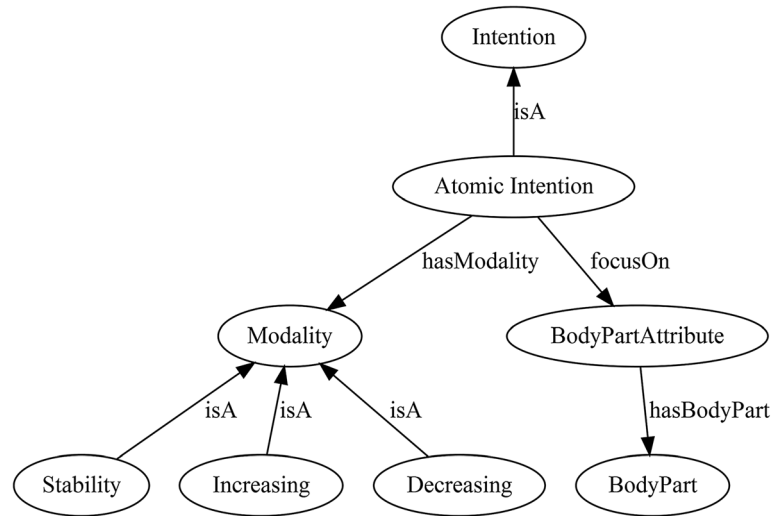


Figure 6 Ontology of atomic intentions.

We illustrate the concepts previously described by an example (see Figure 7): the intention of improving the cardiac activity can be to increase (Modality) the cardiac activity (BodyPartAttribute), which is in turn linked to the BodyPart cardiovascular system. Both Cardiac Activity and Cardiovascular System are individuals of the SNOMED Ontology.

Drug classes ontology

As discussed in Section “GLARE representation language”, in GLARE the representation of a pharmacological action includes, amongst others, an attribute (*substance*) that specifies the drug category (or the specific drug, depending on the level of detail of the CIG itself) prescribed in order to achieve the action’s atomic intention. The value of such an attribute is a link to a sub-concept of the ontological concept DrugCategory (see the higher part of Figure 8).

DrugCategory (and its refinements in the hierarchy) are used in order to achieve (atomic) intentions (see the “aimsTo” arc in Figure 8).

The DrugCategory concept is not flat: it is the root of a hierarchy of categories and its expansion depends on the classification used. In fact, there are in literature several ways to classify the drugs (e.g. those based on the chemical type of the active ingredient, those based on the way they are used to treat a particular condition, etc.). We will use the SNOMED classification for our examples: its concept “Drugs or Medicaments” (that is hierarchically organized) contains our concepts DrugCategory and Drug. However, our approach is independent of the chosen classification, so it is possible to use also other ones (e.g., ATC).

A hierarchical classification of drugs is crucial: several CIGs identify accurate pharmacological prescriptions, pointing out specific drugs, whilst other ones can identify

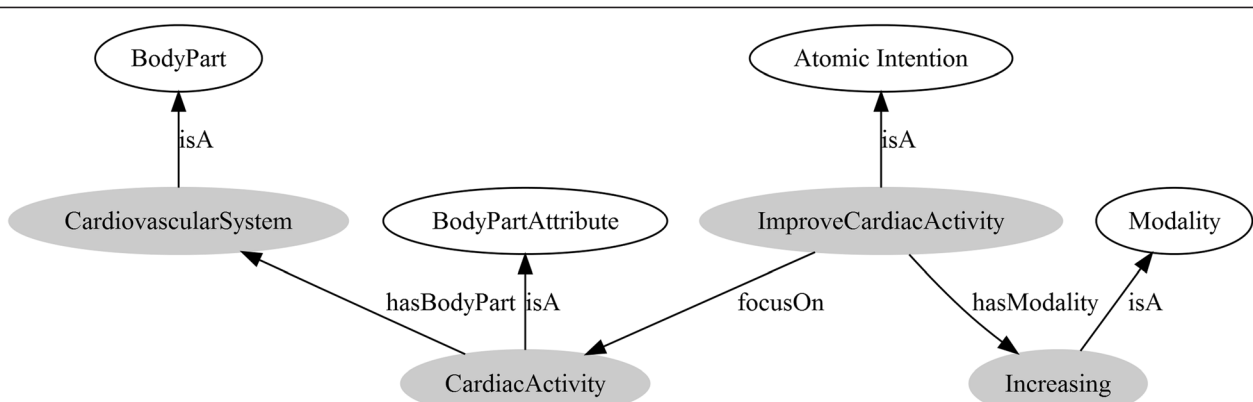
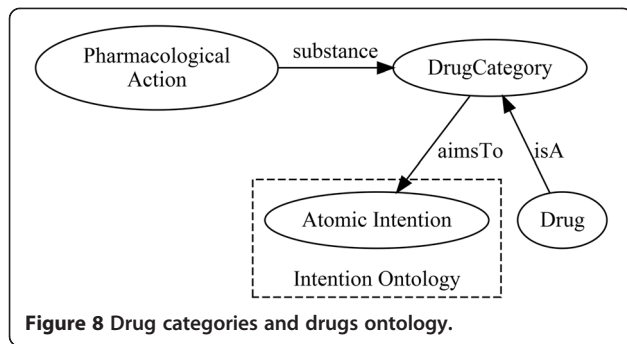


Figure 7 “Improve cardiac activity” intention.



the only drug categories of the pharmacological actions, leaving the physician the task of choosing the drug case by case.

In general, our methodology (see Section “Interaction Detection Algorithm”) is independent of the specific ontology, since it supports the navigation of the drug category ontology along the ISA relationships.

Interactions ontology

As discussed above, our approach supports the description of CIGs at different levels of abstractions. First of all, GLARE supports *top-down action refinement* in the action description, through the decomposition of nested composite actions (plans) into their parts (components). Plans are “high-level” actions, aiming at achieving “high-level” goals, i.e., *intentions* (see Figure 5). At a finer level of details, GLARE pharmacological and work actions (which are part of plans) have *atomic intentions*, which can be achieved through the administration of drugs (see Figure 8). In turn, drugs can be organized through a multiple-level hierarchy of abstraction, going from drug categories to specific drugs (see Figure 9).

Consequently, it is important to notice that interaction can be identified (with different levels of detail) at each one of the different levels of abstraction. As a consequence, in our ontology, we distinguish between IntentionInteractions, AtomicIntentionInteractions, and DrugCategoryInteractions^b. In Figure 10, we show the higher level of our ontology of interactions.

IntentionInteraction is an interaction between two intentions, and has an InteractionType. In Figure 11, we distinguish between three basic types of interactions between intentions. The “Independence” type covers cases

in which the intentions do not interact. The “Concordance” type covers cases in which the two intentions are not independent, but reinforce each other (see example 5 below). The “Discordance” type covers cases in which the two intentions negatively interact with each other (see for instance example 1).

Notice that we have deliberately chosen to support a “fuzzy” representation of interactions. Indeed, in many cases, medical intentions are not 100% in agreement or disagreement. However, more “sharp” types of interactions can also be added to the ontology. For instance, the “Opposite” type could be added as a refinement of the “Discordance” type, to cope with opposite intentions (see example 6 below).

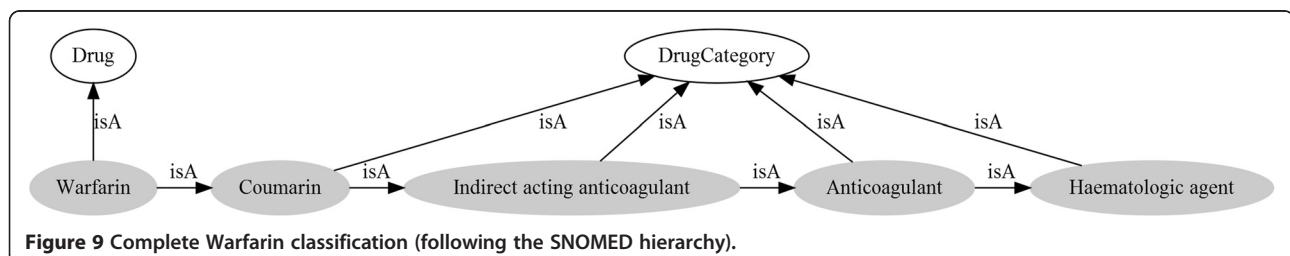
As shown in Figure 11, AtomicIntentionInteraction is a subconcept (ISA arc) of Intention Interaction, being an Interaction between two AtomicIntentions (which, in turn, are Intentions).

Drug category interactions (see Figure 12) occur between two Drug Categories (or subconcepts), and cause a variation of their (atomic) intentions. In our formalism, an interaction between two or more drug categories is modeled as the variation of one or more atomic intentions of the actions involved in the interaction itself, so it is completely described by the intentions it changes and the variation it causes. Since an atomic intention is itself described by a modality, an interaction is a variation of that modality (MV) and its domain is the same set {Increasing, Decreasing, Stability} seen for the concept Modality (see below examples 3, and 7 in the next section).

The representation of drug-drug interactions (i.e., interactions between two Drugs) is similar, but requires at least one of the elements involved belongs to the Drug class (see examples 4, and 8 below).

Interactions detection algorithms

It is worth stressing again that interaction detection should be performed at different levels of abstraction, since, in several cases, top-down refinement along the different hierarchies is needed to detect specific interactions. Of course, results strictly depend on the level of abstraction chosen by the user physician. For instance, some interactions can be discovered considering the higher abstraction level of Intentions.



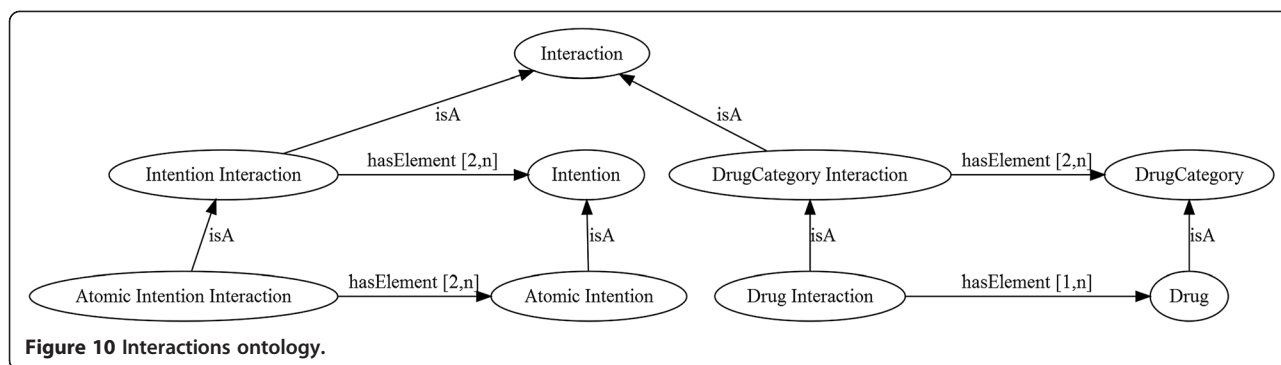


Figure 10 Interactions ontology.

Example 5. Diuretics are a category of drugs frequently used for the treatment of edema (intention). In the case that a patient affected by edema should be treated for hypertension (intention: “decrease blood pressure”), the physician may administer ACE inhibitors or calcium channel blockers. Most of the drugs belonging to these categories do not present explicit interactions with diuretics; however, as shown in Figure 5, the intention of diuretics (i.e., “decrease blood volume” in the Figure 5), of ACE inhibitors (i.e., “inhibition of ACE”) and of calcium channel blockers (i.e., “block of calcium channels”) are all sub intentions (ISA arcs in the figure) of the same high-level intention “decrease blood pressure”. Therefore, the physician can decide to change the posology of the administered drugs.

Other interactions come out only at the lowest abstraction level of Atomic Intentions (e.g., of the pharmacological prescription actions).

Example 6. One of the sub-intentions of the high-level intention to “decrease the urinary retention” is to stimulate contraction of the bladder. That, in turn, can be obtained through the stimulation of the muscarinic receptors (atomic intention), administering, for example, bethanechol^c (drug). If the patient is affected by peptic ulcer a parallel intention should be to decrease the gastric acid secretion, that can be obtained by the inhibition of the muscarinic receptors (atomic intention), administering

pirenzepine^d (drug). The two atomic intentions are obviously conflicting.

In other cases, interactions may show up only at the drug category level: pharmacological actions may have independent intentions, but the drug categories they imply may interact with each other.

Example 7. Vasoconstrictors (drug category), for instance adrenaline (drug), reduce the absorption of the lidocaine (drug), which is a local anesthetic (drug category), enhancing its effect. However, anesthetic intention (for lidocaine) and treatment of anaphylaxis (one of the intentions of adrenaline) do not interact.

Last, some interactions can be detected only at the lowest level, considering specific drugs.

Example 8. Warfarin (drug) is an anticoagulant (drug category) used to prevent the formation of thrombi (intention), whilst erythromycin (drug) is an antibiotic (drug category) commonly used, for instance, in the treatment of respiratory tract infections (intention). There is no interaction between the two categories they belong to; however, the anticoagulant effect of warfarin is enhanced by the simultaneous use of erythromycin.

Our focusing tool is quite innovative, since it allows the physician to navigate both knowledge sources (the CIGs and the ontology) together, thus complementing

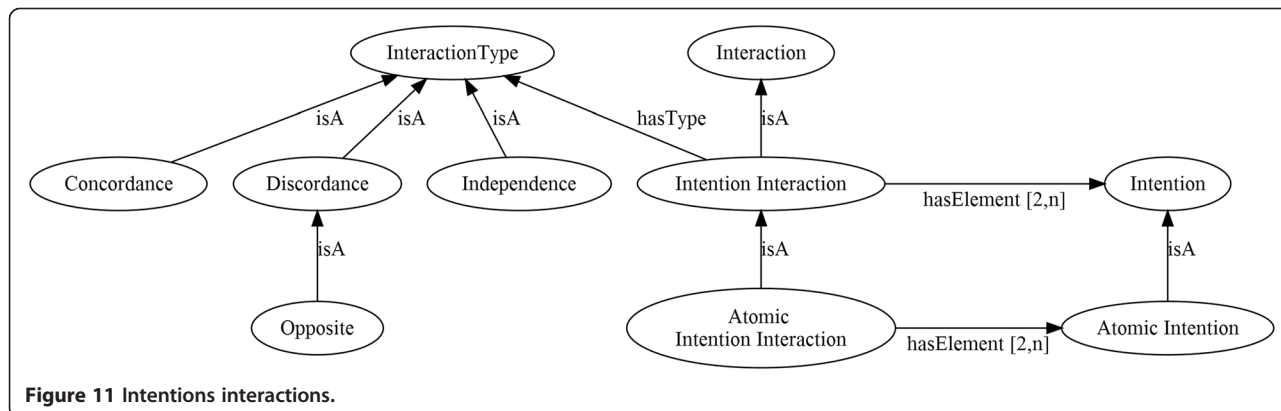


Figure 11 Intentions interactions.

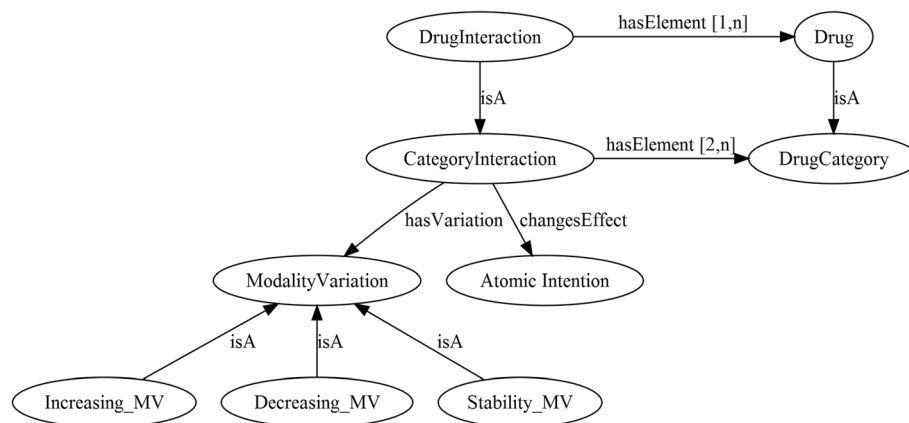


Figure 12 Drug and drug category interactions.

the general knowledge in the CIGs with the additional knowledge in the ontology.

In this section, we will describe a new methodology to explore the above-mentioned knowledge sources in an interactive and flexible way. In Section “Main features of the GLARE system”, we have shown that GLARE (as well as most CIG formalisms) supports different abstraction levels (through the part-of hierarchy of actions) in the description of CPGs. In Section “Interactions: ontological notions”, we have described how our ontology supports the description of different types of interactions at the different abstraction levels, describing not only interactions between general intentions, but also more specific interactions between drug categories, and drugs.

Two main issues must be considered for the definition of an approach for detection of interaction considering the above knowledge sources

- (1) Integration between CIGs and ontology
- (2) Interactions with user-physicians.

Integration between CIGs and ontology

The integration between CIGs and ontology is important, since, in many cases, CIG actions are described at quite a high level of detail, while specific interactions may arise at lower levels. For instance frequently CIG pharmacological actions only indicate drug categories. However, at the very end, a specific drug of that category must be prescribed to the specific patient, and we have previously shown examples in which, while two drug categories do not interact, specific drugs of the two categories do interact. Such interactions can only be detected if the CIG knowledge is *expanded* with the knowledge (about drug interactions) in the ontology, and achieving such an *integration* is one of the main results of our tool.

Abstractly speaking, in our approach, the integration of the knowledge in the CIG and the one in the ontology provides three main levels of detail (henceforth called “dimensions”) at which interactions can be studied:

- (1) **Action dimension.** This dimension is specified directly by the CIG, in which each composite action can be refined into its components (which, in turn, may be composite).
- (2) **Drug category dimension.** Such a dimension is specified by the ontology.
- (3) **Drug dimension.** Such a dimension is specified by the ontology, as the bottom level of the drug category hierarchy.

In our approach, the two knowledge sources (i.e., CIGs and ontology) are strictly connected: as mentioned above, the types of CIG actions and the values of many properties of actions are links to ontological notions (see, e.g., the description of pharmacological actions in Section “GLARE representation language”). In particular, in our approach the representation of pharmacological actions in the CIGs provide the *crucial link for the expansion*: through the *substance* attribute, each pharmacological action specifies a drug category (or, in some cases, a specific drug) to be used, pointing out to the concept representing it in the ontology.

Interactions with user-physicians

In our opinion, a black-box tool that, given two (or more) CIGs and the ontology, provides in output all the possible interaction actions in the CIGs (possibly considering also the possible “expansions” provided by the ontology), is not really useful for the user-physicians. CIGs often consist of hundreds of actions, and the pharmacological prescriptions they contain may often be

achieved through the adoption of several different drugs. Thus, a tool detecting interactions between any pair of actions and drugs belonging to two (or more) CIGs, and blindly operating at all levels of abstraction, would simply output too many interactions to be useful to the physician. For such a reason, we aim at devising a flexible and interactive detection tool allowing physicians to navigate through the different abstraction levels, thus supporting the natural methodology they adopt to cope with comorbidities. For instance, at the highest level, a physician may want to start considering only the interactions between the intentions of the “macro-actions” of the guidelines. Then, focusing on a specific part of the guideline, (s)he may want to move down to a more detailed analysis, considering the decomposition of the “macro action” into its parts, and/or the specific drugs category considered in order to reach the high-level intentions. In general, our approach will provide physicians with the possibility of moving in both directions, i.e., going down from a general to a more specific analysis, or moving up, from a specific analysis to a higher level of abstraction.

Interaction detection process

Summing up, as a result of the above consideration, our approach consists of two main tools:

- (i) a *focusing* tool, allowing physicians to *navigate* into the CIGs to be merged, *selecting* subparts of them and/or expanding/abstracting them at the chosen level of detail (considering all the three dimensions)
- (ii) an *interaction analysis* tool that, given the output of the focusing on two or more CIGs, analyses the interactions, *at the level of abstraction chosen through the focusing tool*.

When analyzing the interaction between two or more CIGs, the user-physician can iteratively use such tools to create her/his “*personalized*” interaction detection process, as required by the specific circumstances and goals. In other words, instead of providing a pre-defined (and unique) way of detecting interactions, and superimpose it on user-physicians, we provide her/him with the proper tools, allowing her/him to *interactively* work out the detection process (s)he needs (please remember the comments to Example in the “Background” Section).

Given the general picture, we now refine it by providing the focusing and the interaction analysis tools.

The focusing tool

The focusing tool is recursive and highly interactive: at each time the physician user chooses a part of one of the guidelines or a specific action and the dimension along which (s)he wants to explore it. Then the system provides the new visualization of the guideline considering

that expansion. The exploration status of each guideline is maintained in a *guideline_navigator*: this data structure keeps track of each guideline expansion done. A guideline navigator can be represented as an ordered list of views (*guideline_view*) of the same CIG: initially, it is initialized to the view that represents the higher level of abstraction of the guideline, then, each time the user performs an exploration action (described in the following), a new level representing the new view is added to the guideline navigator. During the navigation, when the user decides to return to a higher level of abstraction, the last level of the navigator is deleted. Figure 13 shows in detail the structure of a guideline navigator.

Given two or more CIGs to be merged, a guideline navigator is used for each one of them, and the user-physician can operate independently on each one of them. Three basic operations are provided for focusing: ZOOM_IN, ZOOM_OUT, and SELECT.

The SELECT operation allows the user to create a new level of the navigator, which contains only the selected action (s).

The ZOOM-OUT operation recovers the penultimate expansion in a guideline navigator, deleting the last view.

The ZOOM-IN operation creates a new abstraction level in the guideline navigator, allowing for user-driven refinements. The definition of such an operation is quite complex, since it may consider refinements along different and heterogeneous levels of abstraction, and the user can operate recursively (i.e., decide to refine newly determined refinements).

The ZOOM-IN operation is applied on a guideline_navigator (*gn1* in the algorithm in Figure 14), and modifies it, by appending a new guideline_view (*New_View*), which represents the new refinement. At the initialization step, the user has to specify which actions of the last guideline view of *gn1* (s)he wants to modify (*To_Refine_Set* is initialized by such a set of actions). If *To_Refine_Set*

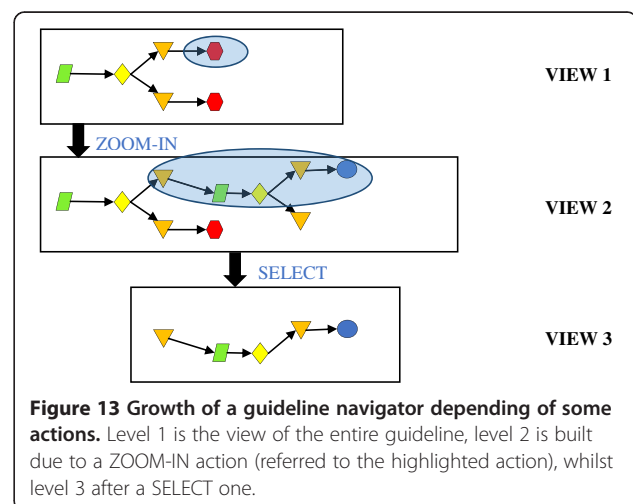


Figure 13 Growth of a guideline navigator depending of some actions. Level 1 is the view of the entire guideline, level 2 is built due to a ZOOM-IN action (referred to the highlighted action), whilst level 3 after a SELECT one.

```
ZOOM_IN(gn1) returns guideline_navigator
  Input: gn1, a guideline navigator

  To_Refine_Set ← user selection of actions from gn1
  if To_Refine_Set ≠ {} then
  begin
    New_View ← generate_view()
    New_View ← copy(Last(gn1))
    Append(New_View, gn1)
    while To_Refine_Set ≠ {} do
    begin
      a ← take an action from To_Refine_Set and remove it from To_Refine_Set
      refinement ← REFINE(New_View, a)
      REPLACE(a, refinement, New_View)
      To_Refine_Set ← To_Refine_Set ∪ user selection of actions from refinement
    end
  end
  return gn1
```

Figure 14 The ZOOM_IN algorithm.

is not empty, then *New_View* is created as a copy of the last view of *gn1*. Then, each action *a* in *To_Refine_Set* is independently refined, through the REFINE operation (see later). The REPLACE operation simply substitutes *a* with the newly identified refinements into *New_View*. Whenever a new refinement is added, we allow users to update *To_Refine_Set*, adding actions of the new refinement to it. In such a way, we provide users with the possibility of going on with this process until the desired level of detail is reached.

The REFINE operation takes in input a guideline_view (*gv*) and *a* action in it, and performs one step of refinement. The way in which such a refinement is obtained depends on the type of *a*. If *a* is a composite action in the guideline_view *gv*, its expansion is simply the sub-guideline constituted by the actions composing it. Thus, the refinement can be easily determined on the basis of the description of the guideline *gv* itself. On the other hand, if *a* is a pharmacological action, the representation of *a* contains the attribute *substance*, whose value is a link to the ontological entity representing drug category to be administered (*drug_cat*). Thus, through the *Down* operator, the algorithm determines the set of descendants of *drug_cat* in the ISA hierarchy in the ontology (the case in which *a* is a specific drug is a degenerate one, in which the result of *Down* is empty). Of course, each one of the descendants c_1, \dots, c_n is a possible alternative refinement of *drug_cat*. To convert this notion into a guideline-like format, the function *BUILD_ALTERNATIVES* constructs a new piece of guideline, consisting of a decision node connected to *n* pharmacological action nodes (each one prescribing one of the drug categories c_1, \dots, c_n). We worked at an OWL implementation of the ontology queries that are part of the algorithms,

thus we highlight them in the figures. We express queries through the Manchester syntax (<http://www.w3.org/2007/OWL/wiki/ManchesterSyntax>) for OWL-DL, supported by Protégé reasoners. In the Figure 15, for example, the descendants of the substance of the action *a* are obtained through the query:

“(Drug or DrugCategory) and isA some (inverse substance value *a*)”.

The nested query “inverse substance value *a*” retrieves all the drugs or drug categories linked to the action *a*, while the external one returns all the elements in the ontology that are directly descendants of at least one of the first (in this case, inheritance is not considered along *isA* arcs).

Interaction analysis algorithm

Once the level of abstraction of the analysis has been specified by the user-physician, through the focusing tool, (s)he may look for the interactions at the *chosen level*. Though the Interaction analysis easily applies also to cases in which more than two CIGs are considered, for the sake of brevity here we consider only the case of two CIGs. After the focusing analysis, the user-physician has determined exactly which parts of the CIGs (s)he is interested in, possibly expanding CIGs with the ontology (e.g., expanding a pharmacological action prescribing a drug category into a therapeutic decision followed by the actions of administering its sub-categories), to reach the desired level of abstraction/detail. At this point, the interaction analysis tool operates as described in the ANALYSE_INTERACTIONS operation below (see Figure 16).

Given two guideline views *gv1* and *gv2*, all composite, work and pharmacological actions are considered. The


```

REFINE (gv, a) returns guideline_view
  Input: gv, a guideline view
         a an action in gv

case a of
  a is a composite action in gv:
    refinement ← expansion of a into its components, considering the guideline
  a is a pharmacological action in gv:
    {c1, ..., cn} ← (Drug or DrugCategory) and
                    isA some (inverse substance value a)
    refinement ← BUILD_ALTERNATIVES({c1, ..., cn})
end case of
return refinement

```

Figure 15 The REFINE algorithm.

set of pairs of actions $\langle a1, a2 \rangle$ such that $a1$ belongs to $gv1$, and $a2$ belongs to $gv2$ is constructed ($intPairSet$), and the result ($intTripleSet$) is initialized to the empty set. Notice that $intTripleSet$ is a set of triples $\langle a1, a2, interactions \rangle$, where $\langle a1, a2 \rangle$ is a pair in $intPairSet$, and $interactions$ is a set of interactions (the possible interactions between $a1$ and $a2$). Then, for each pair $\langle a1, a2 \rangle$ in $intPairSet$, the algorithms of interaction search are applied, returning the set $interactions$ of possible interactions (see the algorithms below), and the triple $\langle a1, a2, interactions \rangle$ is added to the set $intTripleSet$.

Interaction search algorithms operate only on pairs of “homogeneous” actions. Indeed, this is not restrictive, since we provide user with operators (ZOOM-IN,

ZOOM-OUT) to choose the desired level of abstraction. If $a1$ and $a2$ are both composite or work actions, the interactions between their intentions have to be analyzed in the ontology. In our ontology, pharmacological actions have intentions (indeed, atomic intentions), and prescribe the administration of drug categories (or drugs). Thus, both the interactions between intentions and those between drug categories (or drugs) are considered. However, in the case of pharmacological actions added to the CIG by the expansion step (see *BUILD_ALTERNATIVES* in the REFINE operator), no intention is specified (the system cannot automatically determine action intentions), so that only the interactions between the prescribed drug categories (drugs) are considered.

```

ANALYSE_INTERACTIONS (gv1, gv2) returns set of action interactions
  Input: gv1, gv2: guideline views

actSet1 ← set of all the composite, work and pharmacological actions in gv1
actSet2 ← set of all the composite, work and pharmacological actions in gv2
intPairSet ← actSet1 × actSet2
intTripleSet ← {}
for each <a1, a2> in intPairSet do
  begin
    interactions ← {}
    case a1, a2 of
      a1, a2 are both composite or work actions
        interactions ← FIND_INTENTION_INTERACTIONS (a1, a2)
      a1, a2 are both pharmacological actions in the original CIGs
        interactions ← FIND_INTENTION_INTERACTIONS (a1, a2) ∪
                      FIND_DRUG_INTERACTIONS (a1, a2)
      a1, a2 are pharmacological actions, but one of them (or both) are the result of
      an expansion
        interactions ← FIND_DRUG_INTERACTIONS (a1, a2)
    end case of
    intTripleSet ← intTripleSet ∪ {<a1, a2, interactions>}
  end
return intTripleSet

```

Figure 16 The ANALYSE_INTERACTIONS algorithm.

Given two actions $a1$ and $a2$, *FIND_INTENTION_INTERACTIONS* finds the interactions between their intentions. Since actions in CIGs may have multiple intentions, all of them are retrieved (considering the *aimsTo* arcs exiting $a1$ and $a2$ in the ontology). Two types of intention interactions are captured by the algorithm *FIND_INTENTION_INTERACTIONS* in the following. The first type (*onto_interactions* in the algorithm) is determined by looking at the ontological information. For each pair $\langle i1, i2 \rangle$ of intentions (such that $i1$ is an intention of $a1$ and $i2$ of $a2$), the ontology is explored (by following backward the arc *hasElement* of the concept *IntentionInteraction*) in order to determine the interaction holding between them. Then, the retrieved interaction is added to the set *onto_interactions*. On the other hand, some interactions between atomic intentions can be automatically inferred even if they are not directly specified in the ontology. Specifically, if two intentions concern the same *BodyPartAttribute*, but involve different variations for it, the respective interaction can be automatically inferred. Such a second type of interactions is determined by the second part of algorithm. In particular, if two intentions share the same value of *Modality*, their interaction is classified as “Concordance”, whilst, in other cases, its classification is “Discordance”, with a specific case in which the modalities are opposite (“Decreasing” and “Increasing”) and the interaction type is “Opposite”. Such interactions are added to the ontology, and the union of the two types of interactions is returned as output.

Finally, *FIND_DRUG_INTERACTIONS* is defined in a similar way but, obviously, involves searching the ontology for different paths of arcs.

For both the algorithms, we provide an implementation based on OWL. As previously exposed, the *FIND_INTENTION_INTERACTIONS* needs to retrieve two types of interactions: those that already exist into the knowledge base and those that can be automatically inferred starting from it. The first phase of the algorithm (see Figure 17) consists in a query that retrieves all the *IntentionInteractions* between the set of intentions of the first action $a1$ and the one of $a2$. The resulting set of interactions (that corresponds to the *onto_interactions*) is not exhaustive because the ontology does not contain the interactions between all the possible couples of intentions. At this point we need to infer the interaction type between each couple of intentions $\langle i1, i2 \rangle$, belonging respectively to $a1$ and $a2$ sets, that have not a corresponding interaction in *onto_interactions*. In order to achieve this objective we extended the ontology through a set of Semantic Web Rules (<http://www.w3.org/Submission/SWRL/>) and we use them in order to assign to each interaction an inferred type. An interaction regarding $i1$ and $i2$ is added to the ontology (in the algorithm, for the sake of brevity, we denote as ‘new *IntentionInteraction*’ a constructor that creates the individual and the arcs *hasElement* to $i1$ and $i2$), then rules added an arc *hasType* if they recognize automatically a type for that interaction. Rules can automatically recognize both “Concordance” and “Discordance” types (included the “Opposite” one), also considering

```

FIND_INTENTION_INTERACTIONS ( $a1, a2$ ) returns set of interactions
    Input:  $a1, a2$ : actions

    onto_interactions  $\leftarrow$  IntentionInteraction and
        hasElement some (inverse aimsTo value  $a1$ ) and
        hasElement some (inverse aimsTo value  $a2$ )

    for each  $\langle i1, i2 \rangle, i1 \in a1, i2 \in a2$ , that has not an interaction in onto_interactions do
    begin
         $inter \leftarrow$  new IntentionInteraction( $i1, i2$ )
        add  $inter$  to the ontology
    end

    interactions  $\leftarrow$  IntentionInteraction and
        hasElement some (inverse aimsTo value  $a1$ ) and
        hasElement some (inverse aimsTo value  $a2$ )

    return interactions

FIND_DRUG_INTERACTIONS ( $a1, a2$ ) returns set of interactions
    Input:  $a1, a2$ : actions
        interactions: set of interactions

    interactions  $\leftarrow$  (DrugInteraction or CategoryInteraction) and
        hasElement some (inverse substance value  $a1$ ) and
        hasElement some (inverse substance value  $a2$ )

    return interactions
    
```

Figure 17 The *FIND_INTENTION_INTERACTIONS* and the *FIND_DRUG_INTERACTIONS* algorithms.

inheritance of *BodyPartAttributes*. For example, the following couple of rules infer a “Discordance” type considering inheritance:

```
IntentionInteraction(?inter), hasElement(?inter,?i1), hasElement(?inter,?i2), focusOn(?i1,?bpa), focusOn(?i2,?bpa2), isA(?bpa2,?bpa), hasModality(?i1,?mod1), hasModality(?i2,?mod2), Different-From(?mod1,?mod2)  
-> hasType(?inter, discordance)
```

The above rule creates an arc *hasType* that reaches ‘Discordance’, starting from an *IntentionInteraction* *inter* that has two elements *i1* and *i2* that focus on two *BodyPartAttributes* which are, in the *isA* hierarchy, one an ancestor of the other and which have different modalities. For the sake of brevity, we omit other similar rules, such as the one for inferring ‘Concordance’. Due to the Open World Assumption intrinsic in OWL reasoners, the ‘Independence’ type cannot be automatically inferred, but we consider independent all those interactions that have not an interaction type after the application of the above rules.

The last step of the algorithm consist in a second query on the extended ontology. This query is equal to the first one. However, at this time the resulting set of interactions contains also the interactions added in the second part of the algorithm.

The algorithm *FIND_DRUG_INTERACTIONS* (Figure 17) is simpler, because the interactions between drugs are explicitly stored in the ontology. It consists of a query that retrieves all the drug category interactions or drug interactions that involve at least one drug (or drug category) recommended by the action *a1* and one recommended by *a2* (following forward the *substance* arcs).

Results and discussions

Impact on physicians, patients and managers

In this Section, we discuss the impact of CPGs and CIGs in general (with specific reference to the facilities provided by the GLARE system), and then, more specifically, the impact of the approach described in this paper. We consider two different but related issues:

- (1) The analysis of who can *use* CIGs, and our approach, and how/when
- (2) The analysis of the *advantages* that the use of CIGs provides to physicians, patients, and managers/healthcare organizations.

Use of CIG tools

Obviously, the main *users* of CPGs, and of their computerized versions (CIGs), are *physicians*. Indeed, the primary goal of CPGs is to support physicians in their

decision making activities, providing them with appropriate evidence-based recommendations for the diagnosis and/or treatment of a specific disease. CIG approaches further enhance the support to physicians, e.g., automatically focusing on the specific part of the clinical guideline under actual consideration, and automatically “customizing” the guideline to the specific patient at hand, thanks to an automatic connection to the Patient Health Record. Several advanced CIG approaches can be used by physicians to have different forms of decision support. For instance, the approach in Terenziani et al. (2002) provides physicians with “What If” analysis, and the methodology in (Montani and Terenziani 2006; Anselma et al. 2011) applies standard Decision Theory concepts to the CPG context, in order to provide to user-physicians a cost/benefit analysis (where costs are estimated as Quality Adjusted Life Years (Gold et al. 1996)). The treatment of several diseases (e.g., many chronic diseases) involves several actors (e.g., caregivers, family and hospital physicians, community physicians) not geographically co-located. In such a case, CIG approaches like the one in Bottrighi et al. (2013) can provide crucial advantages for agent coordination and delegation, granting for the continuity and the quality of the treatment, as well as for its optimization.

Interestingly, also *patients* may be direct users of some CIG tool. The recent European project *MobiGuide* (www.mobiguide-project.eu/) aims to provide personalized secure clinical-guideline-based guidance to monitored patients also outside the clinical environment. In such a project, users are provided with software tools helping them to personalize certain cost/utility decisions, and to directly access CIGs recommendations.

On the other hand, the healthcare *managers’* perspective is quite different: they are not generally involved in the application of a specific CIG to a specific patient. However, they are focused on clinical processes, on their quality, and on their optimization. Thus, they can use the *consultation facilities* provided by CIG approaches in order to have a general view of how clinical processes should be organized, according to the best-practice derived from the medical evidence. In addition, managers can use approaches such as Bottrighi et al. (2012) in order to check the adherence (*conformance*) of the clinical processes executed in one or more healthcare organizations with the best-practice processes suggested by CIGs. This analysis might be helpful not only to study the quality of the treatments provided, but also for cost/process optimization purposes. Finally, and more important, healthcare managers are involved in the definition and application of healthcare processes in their organizations. Usually, CPGs are “general” bodies of knowledge, often developed by national/international committees, providing “high-level” best-practice recommendations and procedures. Thus, there is often quite a huge gap between

general CPGs (and CIGs) and the actual procedures executed in a specific organization (e.g., in a specific hospital). Both local restrictions (e.g., not all the technological instruments are available in all the hospitals) and local optimizations must be considered, to *contextualize* a general CIG to a specific organization. So far few software tools have been developed to help managers in such a challenging task (consider, e.g., Terenziani et al. (2002)).

General advantages provided by the adoption of CIGs

The adoption of CPGs and, in particular, of CIGs, provides crucial **advantages**. CIGs provide *decision support* to **physicians**, giving them evidence-based recommendations on the treatment of a specific patient, considering patient's data (CIG tools usually support an automatic retrieval of patient's relevant data from the Patient Health Record).

Of course, the main gain is for individual **patients**, since CIGs enforce the quality of healthcare treatments, on the basis of the up-to-date medical evidence.

However, a major gain is also provided to **healthcare managers** and **organizations**. As a matter of fact, the adoption of CIG in a (local, regional, national, or even international) healthcare organization provides crucial advantages such as

- A guarantee about the *quality* of the provided healthcare services (since they are dictated by the most up-to-date medical knowledge and evidence)
- A guarantee about the *standardization* of such services. This issue may be crucial in several geographically “sparse” and not-homogeneous regions, or countries.
- A support in the *coordination* of healthcare services (for treatments that involve multiple and not geographically co-located actors)
- A support for *process* and *cost optimization* (still granting the quality of services).

However, it is important to remember that CPGs are, roughly speaking, a set of *evidence*-based recommendations about the treatment of patients affected by *one* specific disease. Thus, they apply to “typical” patients, affected by a disease. Specific patients, with specific (and atypical, from the point of view of the treated disease) conditions, may not be suited for the treatments suggested by the evidence-based CPG. In such cases, it is usually up to the physician to understand that the patient is, somehow, atypical for the CPG, and to identify the appropriate adaptations of the treatment suggested by the CPG.

The identification of the adaptations is particularly challenging for co-morbid patients, who, having two or more diseases, are “atypical” (with respect to the CPGs coping with such diseases, independently of each others).

This issue is addressed by the approach presented in this paper.

Advantages and uses of our approach for interaction detection

The approach we propose in this paper is a first step in the direction of providing appropriate decision support facilities also in this challenging (but, unfortunately, quite frequently, especially in elder patients) context. In particular, since for co-morbid patients two (or more) CIGs have to be “merged”, we devised a methodology to detect and analyze the interactions between CIGs. The basic and original idea of our approach is that *multiple levels of abstractions* should be used to accomplish such a challenging task. Indeed, operating at different level of abstraction provides at least two main types of advantages to our approach.

- (1) User-friendliness. As a matter of fact, clinical reasoning usually takes into account different levels of detail. First, general tasks are identified, and then refined (top-down refinement). Also, in critical cases, problems in the achievement of low-level tasks can lead back to the revision of higher-level tasks (bottom-up revision process). A user-friendly tool supporting clinical decision must thus provide such a flexibility.
- (2) Generality. Since we support switching among different levels of abstraction, our approach is quite general, and can be exploited in different contexts, and by different users.

In particular, both physicians and managers can use the approach described in this paper.

First of all, our approach can support a **physician** treating a *specific comorbid patient*. In such a context, the possibility of moving down from the “general” actions in the CIGs to the analysis of the interactions of specific drug categories and also of specific drugs is of paramount importance.

However, thanks to its generality, our approach can be used also for the “**abstract**” **analysis** (i.e., just considering the guidelines, with no reference to a specific patient) of the interactions between two or more guidelines that are commonly used together, with the final goal of providing some combined guideline (or, at least, part of it) merging them. In such a case, the possibility of reasoning also about “high-level” actions/tasks in CIGs is certainly crucial.

The analysis of interactions and the “partial merge” of two CIGs may be carried on by **managers** and **physicians**, and may lead to new clinical procedures that ensure both clinical quality and optimization of costs. The detection and analysis of interactions is a crucial step in order to achieve such a goal:

- “negative” interactions (e.g., interactions in which one drug decreases the effect of the other, or, even worse, has the opposite effect of the other) are (usually) to be avoided, since they may be dangerous for the patient and, anyway, are not cost effective.
- “positive” interactions (e.g., interactions in which one drug decreases the effect of the other, or has the same effect of the other) are (usually) desirable, because they achieve the “common” goals of the two guidelines. Of course, if two actions, one in a guideline and one in the other, achieve the same goal and have similar clinical side effects, one (the most costly) of the two may be omitted in the “combination of guidelines”, to optimize costs. Moreover, in the cases of administration of drugs “reinforcing” each other’s effects, the drug posology may be reduced, thus also reducing the cost of treatments.

In general, we strongly believe that, giving the increasing number of co-morbid patients (also due to the increasing aging of the population in many countries), and the social impact of co-morbidities, the definition of new clinical patterns which are both evidence-based and optimized is likely to have a significant positive impact in local and/or national/international healthcare systems and organizations.

Related works

Given its economic and social impact, healthcare management plays a crucial role in modern society. For instance, evidence continues to mount that healthcare is increasingly challenged by entrenched inefficiencies, including wasting more than US\$2 trillion annually Korsten and Seider (2010). As a consequence, several research areas, including computer science, are devoting increasing attention to healthcare.

In particular, given the need to deal with the large amounts of data (“big” data), decision analytics methodologies play an important role in this context.

“Healthcare organizations around the world are challenged by pressures to reduce costs, improve coordination and outcomes, provide more with less and be more patient centric. Yet, at the same time, evidence is mounting that the industry is increasingly challenged by entrenched inefficiencies and suboptimal clinical outcomes. Building analytics competency can help these organizations harness “big data” to create actionable insights, set their future vision, improve outcomes and reduce time to value.” Cortada et al. (2012).

Given the variety of phenomena to be taken into account, several different mainstreams of research have been pursued:

Predictive modeling. The goal of predictive modeling is to derive models that can use patient-specific information to predict the outcome of interest and to thereby support clinical decision-making (Bellazzi and Zupan 2008). Predictive modeling techniques are used, e.g., to predict patient long-term status Zupan et al. (2001), for prognostic and diagnostic classification (Schwarzer et al. 2000).

Data mining. The data mining techniques aim at extracting knowledge from huge amounts of data through automatic or semi-automatic methods:

- Text mining, or text analytics, is the process that extracts relevant information from (big amount of) text expressed in the form of natural language (Chen et al. 2005). It is employed in healthcare, for example, in order to discover new knowledge patterns or hypotheses from the large amount of existing and new literature in biomedicine (Yandell and Majoros 2002). The IBM decision support system Watson (http://www-03.ibm.com/innovation/us/watson/watson_in_healthcare.shtml), based also on natural language processing, “*can incorporate treatment guidelines, electronic medical record data, doctor’s and nurse’s notes, research, clinical studies, journal articles, and patient information into the data available for analysis*”.
- Process mining is an emerging technology in the context of Business Process Management with the goal to derive process models from observed system behavior (Lang et al. 2008). A relevant example is the process mining adopted in the treatment of stroke (Mans et al. 2008).

Evaluation. This family of analytic techniques aims to compare two or more alternative courses of action in terms of costs and outcomes. In particular:

- Cost-benefit analysis is “*a form of economic evaluation that can be used to assess value in terms of money for healthcare intervention*” (Kattan 2009). Managers usually use it in order to evaluate the impact of certain innovations in the health care field. In Wang et al. (2013), e.g., it is used to evaluate the impact of electronic medical records in primary care.
- Cost-effectiveness analysis “*involves comparisons of the additional costs and health benefit of an intervention with those of the available alternatives*” (Kattan 2009). Managers can use it in order to perform resource-allocation decisions considering the ratio of net health-care costs to net health benefits (Weinstein and Stason 1977; Robinson 1993).
- Cost–consequences analysis performs the same task of the cost-effectiveness one, but it does not

aggregate the results (it expresses explicitly, in a table, cost and outcomes of all the alternatives). It is generally more suitable for user physicians in the decision-making process (Mauskopf et al. 1998).

- Risk-benefit analysis weighs the potential for undesirable outcomes and side effects against the potential for positive outcomes of a treatment and is an integral part of the process of determining medical necessity in the delivery of quality medical care. It is particularly used by physicians in order to decide whether to perform a certain treatment on a patient in the presence of known risks (see, e.g., Speroff et al. (1991)).

Neural networks. Despite the fact that neural networks are a technique used in many of the previous fields, their use in medicine has been so extensive that they deserve to be considered as a separate topic. A relevant example of the use of neural networks in healthcare is in the area of decision support (Lisboa and Taktak 2006).

In particular, the approach presented in this paper focuses on a relevant healthcare problem: the treatment of co-morbidities. As explained in subsection “A critical issue: the treatment of co-morbid patients” (in the Background), this is quite a peculiar context, in that the “best practice” profiles of clinical treatment are often available, in the form of CIGs, and the problem is to “compose” two (or more) of them, also on the basis of other forms of explicit knowledge (e.g., knowledge about drug effects and interactions).

Excluding few common features, the existing methods are very different from each other. Sánchez-Garzón et al. (2013), for example, attempts to capture the collaborative aspect of the merging: each guideline is considered by a physician who is expert in the treatment of a single disease, and 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. Another interesting approach, presented in Michalowski et al. (2013) and Willk et al. (2013), uses constraint logic programming to identify and address adverse interactions. In this solution, a constraint logic programming (CLP) model is derived from the combination of logical models that represent each CIG, then a mitigation algorithm is applied to detect and mitigate interactions. Among rule-based systems, López-Vallverdú et al. (2012) represent guidelines as sets of clinical actions that are modeled into an ontology. To combine two treatments, first they are unified in a unique treatment, then a set of “combination rules” is applied to detect and avoid possible interactions. Jafarpour and Abidi (2013) use semantic-web rules and an ontology for the merging criteria. Given these, an Execution Engine dynamically merges several CIGs according to merge criteria. GLINDA proposes a wide ontology of cross-

guideline interactions (Musen et al. 2011). Despite the methods used are very different from each other, one can classify them according to two (orthogonal) criteria. First, CIG can be merged before or during the execution (Abidi 2008). In the second case the methodology “*take account also of execution and of patient data*”, while in the first the result of merging is a general treatment that needs to be customized for each patient and situation. Another important distinction arises between autonomous methods and others that require (or allow) physician intervention. In the first, given two or more CIGs, the system returns a merged guideline without user intervention. In the second, during the process, (s)he can be consulted to take decisions about merging options. This last category (to which, for example, belongs López-Vallverdú et al. (2012)) is very interesting, because it allows the use of a basic medical knowledge that only the physician possesses, and that is very difficult to a priori extract and model into an autonomous system.

Conclusions

The treatment of patients affected by multiple diseases (*comorbid* patients) is one of the main challenges for the modern health care, also due to population aging and to the increase of chronic diseases. In many cases, guidelines coping with each one of the diseases (independently of the others) may be available. However, combining guidelines is a challenging task and a hot topic in the most recent research in Medical Informatics. Though several valuable proposals have been pointed out in the specialized literature (see the “Related Works” section), the complexity of the overall problem demands additional steps forward of the state-of-the-art.

In this paper, we address a significant part of the problem, namely the detection of interactions between CIGs. Relevant studies demonstrate that various types of interactions must be taken into account when merging two (or more) CIGs, and propose an ontology of interactions (Musen et al. 2011). However, to the best of our knowledge, our approach is the first one that identifies different levels of abstractions in the analysis of interactions, based on both the hierarchical organization of CIGs (in which composite actions are refined into their components) and the hierarchy of drug categories (specific drugs often constitute the bottom of such a hierarchy). We show that interactions can occur at each level of abstraction, so that the medical interaction detection process must navigate all levels. In addition, medical experience shows that the detection of interaction must be an *interactive* process, in which, at each step, user-physicians decide how to proceed on the basis of the previous steps.

In this paper, we propose a general methodology (data/knowledge structures and reasoning algorithms

operating on them) supporting user-driven and interactive interaction detection over multiple levels of abstraction. To the best of our knowledge, no other approach in the literature has explicitly addressed such a challenging goal. In this sense, we believe that our approach, besides being innovative, is somehow *complementary* in respect of several other approaches in the literature, so that an integration with them can be a possible area of future work (e.g., with Riaño and Collado (2013) methodology to merge CIGs).

The main contributions of our work are:

- (1) We identified three different abstraction levels at which interactions can occur: between actions in CIGs, between drug categories, and, finally, between specific drugs
- (2) We worked out an ontology of interactions, considering the three levels above
- (3) We clarified that the study of CIG interactions can be performed only by considering the expansion of CIGs with (part of) the ontological knowledge
- (4) Last, but most important, we proposed a set of tools and operations supporting *user-physician-driven interactive* interaction detection along the three different *levels of abstraction*.

The above results can be regarded as the starting point of a new computer-based methodology to cope with comorbidities, based on the notion of multiple abstractions. Carrying on such a new line of research will involve interesting new scientific challenges, especially concerning the development of suitable *flexible* and *interactive* approaches to support physicians in the *solution of interactions at different levels of abstractions*.

We are currently developing a prototypical implementation to demonstrate our approach, based on GLARE. In our short-term future work, we aim at extending our approach to cope also with “patient – CIG action” interactions and “patient – drug” interactions, and to cope with the temporal issues (e.g., not all interactions between CIGs are possible, due to the temporal constraints between CIG actions).

In our long-term future work, we will attempt to achieve the goals specified in the section “Goals of our overall approach”: supporting physicians also in the interaction solving, and, finally, in merging multiple guidelines in the treatment of the specific patient at hand. In order to achieve such a challenging goal, we envision extending our current approach to decision-theory-based cost/benefit analysis. As discussed in the subsection “Decision-theory-based decision support in GLARE”, already adopts cost/benefit analysis to support the choice between clinically equivalent therapies. However, not only in our approach Montani and Terenziani

(2006), but in most healthcare analytics techniques, only single morbidities are taken into account. Our long-term goal is to integrate our Decision Support System with the knowledge needed to merge the results of single-morbidity analytics in order to provide decision-makers with all those decision support facilities already investigated and developed considering single diseases only.

Endnotes

^aPlease notice that, in this paper, we deliberately use the term “intention” in a broad sense, to cover not only the desired effects of medical actions, but also the effects of pharmacological prescriptions, and of drugs.

^bIndeed, in our ontology we also take into account other types of interactions, involving the status of the patient. In particular, the patient status may interact with the Intentions of composite actions, with the atomic intentions of pharmacological prescriptions, and with DrugCategories. Such issues are not taken into account in this paper, where we focus only on “patient-independent” detection of CIGs interactions.

^cBethanechol is an agonist of muscarinic receptors, belonging to the acetylcholine drug category.

^dPirenzepine is an antagonist of muscarinic receptors, belonging to the atropine drug category.

^eThe keyword *inverse* is used to navigate forward a link. In particular, the query returns all the elements in the co-domain that are connected to the element (s) in the given domain (defined through the *value* construct). On the contrary, when *inverse* is absent, properties are navigated backwards.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

LP, GM and PT have strictly cooperated on all the different aspects discussed in the paper. In particular, GM has provided the medical knowledge, while PT and LP developed the algorithmic part. All authors read and approved the final manuscript.

Acknowledgements

The authors are very grateful to Yuval Shahar, for his invaluable advise and for many inspiring suggestions. The work described in paper was partially supported by Compagnia di San Paolo, in the Ginseng project.

Author details

¹Department of Computer Science, Università degli Studi di Torino, C.so Svizzera, 185, Turin 10135, Italy. ²Azienda Ospedaliera San Giovanni Battista, Turin, Italy. ³DISIT, Università del Piemonte Orientale, Viale Teresa Michel 11, Alessandria 15121, Italy.

Received: 3 April 2014 Accepted: 15 April 2014

Published: 2 May 2014

References

- Abidi, SR. (2008). A conceptual framework for ontology based automating and merging of clinical pathways of comorbidities. *K4HelP*, 5626, 55–66.
- Anselma, L, Terenziani, P, Montani, S, & Bottrighi, A. (2006). Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. *Artificial Intelligence in Medicine*, 38(2), 171–195.

- Anselma, L, Bottrighi, A, Molino, G, Montani, S, Terenziani, P, & Torchio, M. (2011). Supporting knowledge-based decision making in the medical context: The glare approach. *IJKBO*, 1(1), 42–60.
- Anselma, L, Bottrighi, A, Montani, S, & Terenziani, P. (2013). Extending bcdm to cope with proposals and evaluations of updates. *IEEE Transactions on Knowledge and Data Engineering*, 25(3), 556–570.
- Bellazzi, R, & Zupan, B. (2008). Predictive data mining in clinical medicine: current issues and guidelines. *International Journal of Medical Informatics*, 77(2), 81–97.
- Bottrighi, A, Giordano, L, Molino, G, Montani, S, Terenziani, P, & Torchio, M. (2010). Adopting model checking techniques for clinical guidelines verification. *Artificial Intelligence in Medicine*, 48(1), 1–19.
- Bottrighi, A, Chesani, F, Mello, P, Montali, M, Montani, S, & Torroni, P. (2012). Conformance checking of executed clinical guidelines in presence of basic medical knowledge. *Business Process Management Workshops*, 100, 200–211.
- Bottrighi, A, Molino, G, Montani, S, Terenziani, P, & Torchio, M. (2013). Supporting a distributed execution of clinical guidelines. *Comput Methods Prog Biomed*, 112(1), 200–210.
- Chen, H, Fuller, S, Friedman, C, & Hersh, W. (2005). Knowledge Management, Data Mining, and Text Mining in Medical Informatics. In *Medical Informatics* (pp. 3–33). Springer: US.
- Cortada, JW, Gordon, D, & Lenihan, B. (2012). *The value of analytics in healthcare*. New York, USA: IBM Global Services.
- Fox, J, Johns, N, & Rahmzadeh, A. (1998). Disseminating medical knowledge: the proforma approach. *Artificial Intelligence in Medicine*, 14(1–2), 157–182.
- Fridsma, DB. (2001). Special issue on workflow management and clinical guidelines. *Journal of the American Medical Informatics Association*, 22(1), 1–80.
- Gijsen, R, Hoeymans, N, & Schellevis, FG. (2001). Causes and consequences of comorbidity: a review. *Journal of Clinical Epidemiology*, 54(7), 661–674.
- Gold, M, Siegel, JE, Russell, LB, & Weinstein, MC. (1996). *Cost-Effectiveness in Health and Medicine*. New York: Oxford University Press.
- Gordon, C, & Christensen, JP. (1995). *Health Telematics for Clinical Guidelines and Protocols*. Amsterdam, Netherlands: IOS Press.
- Gruber, T. (2009). Ontology. In L Liu & M Tamer Özsu (Eds.), *Encyclopedia of Database Systems* (pp. 1963–1965). US: Springer.
- Institute of Medicine, Committee on Quality Health Care in America. (2001). *Crossing the Quality Chasm: A New Health System for the 21st Century* (p. 151). Washington, USA: National Academy Press.
- Jafarpour, B, & Abidi, SS. (2013). *Merging Disease-Specific Clinical Guidelines to Handle Comorbidities in a Clinical Decision Support Setting*. 14th Conference on Artificial Intelligence in Medicine. 7885 (pp. 28–32). Murcia, Spain: Springer.
- Kattan, MW. (2009). *Encyclopedia of Medical Decision Making*. Thousand Oaks, California, USA: SAGE Publications Inc.
- Koppel, R, Metlay, JP, & Cohen, A. (2005). Role of Computerized Physician Order Entry Systems in Facilitating Medication Errors. *JAMA : the journal of the American Medical Association*, 293(10), 1197–1203.
- Korsten, P, & Seider, C. (2010). *The world's 4 trillion dollar challenge: Using a system-of-systems approach to build a smarter planet*. New York, USA: IBM Institute for Business Value.
- Lang, M, Bürkle, T, Laumann, S, & Prokosch, H. (2008). Process mining for clinical workflows: challenges and current limitations. *Studies in health technology and informatics*, 136, 229–234.
- Liaw, ST, Rahimi, A, Ray, P, Taggart, J, Dennis, S, de Lusignan, S, & Talaei-Khoei, A. (2013). Towards an ontology for data quality in integrated chronic disease management: A realist review of the literature. *International journal of medical informatics*, 82(1), 10–24.
- Lisboa, PJ, & Taktak, AF. (2006). The use of artificial neural networks in decision support in cancer: a systematic review. *Neural networks: the official journal of the International Neural Network Society*, 19(4), 408–415.
- López-Vallverdú, JA, Riaño, D, & Collado, A. (2012). Rule-Based Combination of Comorbid Treatments for Chronic Diseases Applied to Hypertension, Diabetes Mellitus and Heart Failure. *ProHealth/KR4HC*, 7738, 30–41.
- Mans, R, Schonenberg, H, Leonardi, G, Panzarasa, S, Cavallini, A, Quaglini, S, & van der Aalst, W. (2008). Process mining techniques: an application to stroke care. *Studies in health technology and informatics*, 136, 573–578.
- Mauskopf, J, Paul, J, Grant, D, & Stergachis, A. (1998). The role of cost-consequence analysis in healthcare decision-making. *Pharmacoeconomics*, 13(3), 277–288.
- McGuinness, DL, & van Harmelen, F. (2004). *OWL Web Ontology Language* (In W3C Recommendation). <http://www.w3.org/TR/owl-features/>.
- Michalowski, M, Wilk, S, Michalowski, W, Farion, K, Lin, D, & Mohapatra, S. (2013). Using Constraint Logic Programming to Implement Iterative Actions and Numerical Measures during Mitigation of Concurrently Applied Clinical Practice Guidelines. In *14th Conference on Artificial Intelligence in Medicine*. Murcia, Spain: Springer.
- Montani, S, & Terenziani, P. (2006). Exploiting decision theory concepts within clinical guideline systems: Toward a general approach. *International Journal of Intelligent Systems*, 21(6), 585–599.
- Musen, M, Goldstein, MK, Tu, S, Martins, S, Nyulas, C, Jung, H, & Kum, P. (2011). *GLINDA Interaction Ontology* (In GLINDA: GuideLine Interaction Detection Architecture). <http://glinda-project.stanford.edu/guidelineinteractionontology.html>.
- Peleg, M. (2013). Computer-interpretable clinical guidelines: a methodological review. *Journal of biomedical informatics*, 46(4), 744–763.
- Peleg, M, Boxwala, AA, & Ogunyemi, O. (2000). GIf3: the evolution of a guideline representation format. *Proceedings/AMIA ... Annual Symposium. AMIA Symposium*, 645–649.
- Pérez-Rey, D, Majojo, V, García-Remesal, M, Alonso-Calvo, R, Billhardt, H, Martín-Sánchez, F, & Sousa, A. (2006). ONTOFUSION: Ontology-based integration of genomic and clinical databases. *Computers in biology and medicine*, 36(7–8), 712–730.
- Quaglini, S, Stefanelli, M, Lanzola, G, Caporusso, V, & Panzarasa, S. (2001). Flexible guideline-based patient care. *Artificial Intelligence in Medicine*, 22(1), 65–80.
- Riaño, D, & Collado, A. (2013). *Model-Based Combination of Treatments for the Management of Chronic Comorbid Patients*. 14th Conference on Artificial Intelligence in Medicine. 7885 (pp. 11–16). Murcia, Spain: Springer.
- Robert Wood Johnson Foundation and Partnership for Solutions. (2004). *Chronic Conditions: Making the Case for Ongoing Care*. Johns Hopkins University. <http://www.partnershipforsolutions.org/DMS/files/chronicbook2004.pdf>. Accessed 17 Apr 2014.
- Robinson, R. (1993). Cost-effectiveness analysis. *BMJ (Clinical research ed.)*, 307(6907), 793–795.
- Rubin, DL, Lewis, SE, Mungall, CJ, Misra, S, Westerfield, M, Ashburner, M, & Musen, MA. (2006). National Center for Biomedical Ontology: advancing biomedicine through structured organization of scientific knowledge. *Omic: a journal of integrative biology*, 10(2), 185–198.
- Sánchez-Garzón, I, Fernández-Olivares, J, Onaindia, E, Milla, G, Jordán, J, & Castejón, P. (2013). A Multi-agent Planning Approach for the Generation of Personalized Treatment Plans of Comorbid Patients. In *14th Conference on Artificial Intelligence in Medicine*, 7885 (pp. 23–27).
- Schwarzer, G, Vach, W, & Schumacher, M. (2000). On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. *Statistics in medicine*, 19(4), 541–561.
- Shahar, Y, Musen, M, & Tu, SW. (1996). Eon: a component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Association: JAMA*, 3(6), 367–388.
- Shahar, Y, Miksch, S, & Johnson, PD. (1998). The Asgaard project: a task-specific framework for the application. *Artificial Intelligence in Medicine*, 14(1–2), 29–51.
- Shiffman, RN, Karras, BT, Agrawal, A, Chen, R, Marengo, LN, & Nath, S. (2000). Model formulation: Gem: a proposal for a more comprehensive guideline document model using xml. *Journal of the American Medical Informatics Association: JAMIA*, 7(5), 488–498.
- Sittig, DF, Wright, A, Osheroff, JA, Middleton, B, Teich, JM, Ash, JS, & Bates, DW. (2008). Grand challenges in clinical decision support. *Journal of biomedical informatics*, 41(2), 387–392.
- Speroff, T, Dawson, N, Speroff, L, & Haber, R. (1991). A risk-benefit analysis of elective bilateral oophorectomy: effect of changes in compliance with estrogen therapy on outcome. *American journal of obstetrics and gynecology*, 164(1), 165–174.
- Stantic, B, Terenziani, P, Governatori, G, Bottrighi, A, & Sattar, A. (2012). An implicit approach to deal with periodically repeated medical data. *Artificial Intelligence in Medicine*, 55(3), 149–162.
- Tatman, JA, & Shachter, R. (1990). Dynamic programming and influence diagrams. *IEEE transactions on systems, man, and cybernetics*, 20(2), 365–379.
- Ten Teije, A, Miksch, S, & Lucas, P. (2008). *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*. Amsterdam, Netherlands: IOS Press.
- Terenziani, P, Molino, G, & Torchio, M. (2001). A modular approach for representing and executing clinical guidelines. *Artificial Intelligence in Medicine*, 23(3), 249–276.
- Terenziani, P, Montani, S, Bottrighi, A, Torchio, M, & Molino, G. (2002). Supporting physicians in taking decisions in clinical guidelines: the GLARE “what if” facility. In *Proceedings of the AMIA Symposium* (p. 772).

- Terenziani, P, Bottrighi, A, Torchio, M, Molino, G, Anselma, L, & Correndo, G. (2003). Applying artificial intelligence to clinical guidelines: the glare approach. (A. Cappelli, & F. Turini, A cura di). *AI*A*, 2829, 536–547.
- Terenziani, P, Montani, S, Bottrighi, A, Molino, G, Torchio, M, & Correndo, G. (2004). A context-adaptable approach to clinical guidelines. *Studies in health technology and informatics*, 107(Pt 1), 169–173.
- Wang, S, Middleton, B, Prosser, L, Bardou, C, Spurr, C, Carchidi, P, & Bates, D. (2013). A cost-benefit analysis of electronic medical records in primary care. *The American journal of medicine*, 114(5), 397–403.
- Weinstein, M, & Stason, W. (1977). Foundations of cost-effectiveness analysis for health and medical practices. *The New England journal of medicine*, 296(13), 716–721.
- Wilk, S, Michalowski, M, Michalowski, W, Farion, K, Hing, MM, & Mohapatra, S. (2013). Mitigation of adverse interactions in pairs of clinical practice guidelines using constraint logic programming. *Journal of biomedical informatics*, 46(2), 341–353.
- Yandell, MD, & Majoros, WH. (2002). Genomics and Natural Language Processing. *Nature reviews. Genetics*, 3(8), 601–610.
- Zupan, B, Demsar, J, Smrke, D, Bozikov, K, Stankovski, V, Bratko, I, & Beck, JR. (2001). Predicting patient's long-term clinical status after hip Arthroplasty using hierarchical decision modelling and data mining. *Methods of information in medicine*, 40(1), 25–31.

doi:10.1186/2193-8636-1-8

Cite this article as: Piovesan et al.: An ontological knowledge and multiple abstraction level decision support system in healthcare. *Decision Analytics* 2014 1:8.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
