Universidade do Minho
Escola de Engenharia

Tiago José Martins Oliveira

Clinical Decision Support: Knowledge Representation and Uncertainty Management

January 2017

Tiago José Martins Oliveira **Clinical Decision Support: Knowledge Representation and Uncertainty Management**

UMinho|2017

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

POPH
QUALIFICAR É CRESCER.

QREN
QUADRO
DE REFERÊNCIA
ESTRATÉGICO
NACIONAL
PORTUGAL 2007.2013

Governo da
República Portuguesa

UNIÃO EUROPEIA
Fundo Social Europeu

Universidade do Minho
Escola de Engenharia

Tiago José Martins Oliveira

Clinical Decision Support: Knowledge Representation and Uncertainty Management

Doctoral Thesis
Doctoral Degree in Biomedical Engineering

Thesis supervised by
Professor Paulo Jorge Freitas de Oliveira Novais
Professor José Carlos Ferreira Maia Neves

January 2017

STATEMENT OF INTEGRITY

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of elaboration of this thesis. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, 02/01/2017

Full name: Tiago José Martins Oliveira

Signature:

Tiago José Martins Oliveira

ACKNOWLEDGEMENTS

"Look up at the stars and not down at your feet. Try to make sense of what you see, and wonder about what makes the universe exist. Be curious."

- Stephen Hawking (2012)

This doctoral thesis was the product of many contributions provided by important people that crossed my path and left their mark in a positive, unforgettable way. This is a big thank you to you all for pushing me to look up at the stars.

Professor Paulo Novais, whom I now call my friend, opened up a new world to me and, for that, I will forever be in his debt. Professor Paulo always offered the guidance I needed, at the professional and personal levels, and gave me the opportunity to work with him at the Intelligent Systems Lab, which was an immense pleasure and an experience that shaped the person I am today.

To Professor José Neves, who always pushed me to do my best, to improve, to be better, I address my gratitude. His experience and advice helped me pursue my goals and, for that, I cannot express how grateful I am.

I would also like to thank Professor César Analide who was always there throughout my stay at the Intelligent Systems Lab with great advice and a friendly word.

Professor Ken Satoh gave me the opportunity to work up close with him and fulfil the lifelong dream of travelling to Japan. I will always treasure the moments I spent at the National Institute of Informatics (NII) and all the (many) things he has taught me.

My thanks to Professor Vicente Julián for welcoming me at the Grupo de Tecnología Informática - Inteligencia Artificial (GTI-IA), at Valencia University of Technology (UPV). I am very grateful to all the members of the lab.

My sincere thanks to the MD Pedro Leão, who kindly helped me to understand the domain and was always available to answer my questions.

A special word of gratitude to my friends from the Intelligent Systems Lab: Ângelo Costa, Davide Carneiro, Marco Gomes, Fábio Silva, João Ramos, André Pimenta, António Silva, Filipa Ferraz, Dalila Durães, and Ana Silva. I am very happy to have met you and to have collaborated with you.

I would like to dedicate this thesis and all the effort behind it, all the the time spent, and all the emotional roller-coaster brought about in its completion to my family and friends. My parents, my grandparents, and my friends mean the world to me and they always support me in whatever I set my mind to do. Thank you for being there for me.

The work of the PhD candidate Tiago José Martins Oliveira is supported by a grant from FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) with the reference SFRH/BD/85291/ 2012.

POPH
QUALIFICAR É CRESCER.

QREN
QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL
PORTUGAL 2007.2013

GOVERNO DA REPÚBLICA PORTUGUESA

UNIÃO EUROPEIA
Fundo Social Europeu

ABSTRACT

Decision-making in clinical practice is faced with many challenges due to the inherent risks of being a health care professional. From medical error to undesired variations in clinical practice, the mitigation of these issues seems to be tightly connected to the adherence to Clinical Practice Guidelines as evidence-based recommendations

The deployment of Clinical Practice Guidelines in computational systems for clinical decision support has the potential to positively impact health care. However, current approaches to Computer-Interpretable Guidelines evidence a set of issues that leave them wanting. These issues are related with the lack of expressiveness of their underlying models, the complexity of knowledge acquisition with their tools, the absence of support to the clinical decision making process, and the style of communication of Clinical Decision Support Systems implementing Computer-Interpretable Guidelines. Such issues pose as obstacles that prevent these systems from showing properties like modularity, flexibility, adaptability, and interactivity. All these properties reflect the concept of *living guidelines*.

The purpose of this doctoral thesis is, thus, to provide a framework that enables the expression of these properties.

The modularity property is conferred by the ontological definition of Computer-Interpretable Guidelines and the assistance in guideline acquisition provided by an editing tool, allowing for the management of multiple knowledge patterns that can be reused. Flexibility is provided by the representation primitives defined in the ontology, meaning that the model is adjustable to guidelines from different categories and specialities.

On to adaptability, this property is conferred by mechanisms of Speculative Computation, which allow the Decision Support System to not only reason with incomplete information but to adapt to changes of state, such as suddenly knowing the missing information.

The solution proposed for interactivity consists in embedding Computer-Interpretable Guideline advice directly into the daily life of health care professionals and provide a set of reminders and notifications that help them to keep track of their tasks and responsibilities.

All these solutions make the CompGuide framework for the expression of Clinical Decision Support Systems based on Computer-Interpretable Guidelines.

RESUMO

A tomada de decisão na prática clínica enfrenta inúmeros desafios devido aos riscos inerentes a ser um profissional de saúde. Desde o erro médico até às variações indesejadas da prática clínica, a atenuação destes problemas parece estar intimamente ligada à adesão a Protocolos Clínicos, uma vez que estes são recomendações baseadas na evidência.

A operacionalização de Protocolos Clínicos em sistemas computacionais para apoio à decisão clínica apresenta o potencial de ter um impacto positivo nos cuidados de saúde. Contudo, as abordagens atuais a Protocolos Clínicos Interpretáveis por Máquinas evidenciam um conjunto de problemas que as deixa a desejar. Estes problemas estão relacionados com a falta de expressividade dos modelos que lhes estão subjacentes, a complexidade da aquisição de conhecimento utilizando as suas ferramentas, a ausência de suporte ao processo de decisão clínica e o estilo de comunicação dos Sistemas de Apoio à Decisão Clínica que implementam Protocolos Clínicos Interpretáveis por Máquinas. Tais problemas constituem obstáculos que impedem estes sistemas de apresentarem propriedades como modularidade, flexibilidade, adaptabilidade e interatividade. Todas estas propriedades refletem o conceito de *living guidelines*.

O propósito desta tese de doutoramento é, portanto, o de fornecer uma estrutura que possibilite a expressão destas propriedades.

A modularidade é conferida pela definição ontológica dos Protocolos Clínicos Interpretáveis por Máquinas e pela assistência na aquisição de protocolos fornecida por uma ferramenta de edição, permitindo assim a gestão de múltiplos padrões de conhecimento que podem ser reutilizados. A flexibilidade é atribuída pelas primitivas de representação definidas na ontologia, o que significa que o modelo é ajustável a protocolos de diferentes categorias e especialidades.

Quanto à adaptabilidade, esta é conferida por mecanismos de Computação Especulativa que permitem ao Sistema de Apoio à Decisão não só raciocinar com informação incompleta, mas também adaptar-se a mudanças de estado, como subitamente tomar conhecimento da informação em falta.

A solução proposta para a interatividade consiste em incorporar as recomendações dos Protocolos Clínicos Interpretáveis por Máquinas diretamente no dia a dia dos profissionais de saúde e fornecer um conjunto de lembretes e notificações que os auxiliam a rastrear as suas tarefas e responsabilidades.

Todas estas soluções constituem a estrutura CompGuide para a expressão de Sistemas de Apoio à Decisão Clínica baseados em Protocolos Clínicos Interpretáveis por Máquinas.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

A

AI - Artificial Intelligence

AIM - Artificial Intelligence in Medicine

C

CBR - Case-based Reasoning

CDSS - Clinical Decision Support System

CIG - Computer-Interpretable Guideline

CPG - Clinical Practice Guideline

E

ES - Expert System

G

GLARE - GuideLine Acquisition, Representation and Execution

GLIF3 - Guideline Interchange Format

GTI-IA - Grupo de Tecnologia Informática - Inteligencia Artificial

N

NII - National Institute of Informatics

O

OWL - Web Ontology Language

S

SAGE - Standards-based Sharable Active Guideline Environment

T

TNM - Task Network Model

U

UPV - Valencia University of Technology

1

INTRODUCTION

Conducting medical practice is, in its essence, making decisions. Health care professionals (typically physicians and nurses) are constantly faced with situations in which there is the need to select one of several options and, at the same time, to predict the possible effects of the selected option on the patient. The range of clinical decisions consists of two large groups (Thompson and Dowding, 2009): diagnosis and management.

Diagnosis refers to the analysis of available data to determine the pathophysiologic explanation for a patient's symptoms. It is a complex process where the health care professional needs to determine which questions to ask, tests to perform, and the values of results. Thus, it is not only necessary to decide what is true about a patient but also what data is needed to decide what is true.

Management is equally demanding. The health care professional has to follow up on the patient, ensuring that his health status evolves positively or, depending on the severity of the situation, does not deteriorate any further. Therefore, it is required of him to decide when to intervene or even the need to intervene. Furthermore, he has to be constantly monitoring the patient and using therapy responses to guide the process, one a step at a time.

This Introduction Section aims to characterize clinical decision-making and refer the challenges that clinical practice is faced with. Since this is a doctoral thesis in Biomedical Engineering with a strong component in Computer Science, the focus is placed on the solutions that the last field has to offer in order to meet those challenges. As such, a brief characterization of Clinical Decision Support Systems (CDSSs) is made, with a reflection on their evolution throughout the years, impact, and identification of the main limitations of CDSSs based on Computer-Interpretable Guidelines (CIGs), which are particularly oriented for the provision of decision support in diverse situations. Based on the identified limitations, the hypothesis and objectives of the doctoral thesis are defined. Additionally a guide for the document is provided.

1.1 CLINICAL DECISION-MAKING

The nature of clinical decision-making is that of a process that can be fast and intuitive or well reasoned, based on heuristics and analytical evidence-based decisions. There is a balance that must be achieved between the intuition and experience of the health care professional, given the high volume of decisions to be made. An observational study conducted by Bucknall (2000) on a medical ward in Australia observed that nurses made a patient care decision every 30 seconds, and another study, this time conducted by Watson (1994) in the United Kingdom, determined that nurses made 18 decisions every 20 minutes for a period of two hours. Of course nurses are an extreme case, given their closer and longer contact with patients. However, these are good indicators of the frequency and velocity of clinical decisions.

According to the Oxford Online Dictionary, a decision is defined as being a *"conclusion or resolution reached after consideration"* (Oxford Online Dictionary, 2016). The work of Matteson and Hawkins (1990) further characterizes a decision by stating that: it ends doubt or debate, it is based on indications or evidence, it involves a deliberate mental choice, and, finally, it includes two or more options. The actual decision is the last point of the decision-making process, after a complex series of deliberations, occurring in an environment of uncertainty (Thompson and Dowding, 2009), as is the case in health care. However, rather than being just an obstacle to the decision-making process, uncertainty is a defining characteristic of a decision. If there were no uncertainty in the relationship between the problem and the outcome, there would not be the need for a decision in the first place, and the best solution to the problem would be immediately calculated.

1.1.1 *The Decision-Making Process*

Typically, the process of making a clinical decision, or any decision for that matter, follows a set of necessary steps, from gathering the necessary information to the outcome. Yet, this should not be seen as a linear process. Ideally one step should inform the other, but emerging information may cause one to jump back and forth, according to its interpretation. These steps can be framed according to numerous models used to interpret and explain the process. In terms of clinical decision-making, there are two antagonistic conceptual frameworks derived from general decision-making. They are the analytical framework and the intuitive framework (Hammond et al., 1987).

The Analytical Framework

In the analytical approach, the health care professional tries to associate the situation he is presented with to a set of guiding principles or rules. This framework is based, to an extent, on the assumptions from information processing theory, according to which the decision maker stores relevant information in his memory and problem solving occurs when he retrieves this information from both long and short-term memory (Han et al., 2007; Hunink et al., 2014). The information gained from education and experience is stored in the long-term memory, which has more capacity, and the decision maker uses the access to short-term memory information to stimulate the retrieval of long-term memories. For instance, when presented with a case, the health care professional will use his short-term memories of the patient's signs and symptoms to retrieve, from his long-term memory, information about the signs and symptoms for the type of diagnosis he is performing.

The information processing theory is the basis for several decision-making models. The one developed by (Carnevali et al., 1993) defines seven stages of clinical decision-making, more oriented to diagnosis. The stages are as follows:

1. Exposure to pre-encounter data;
2. Entry to the data search field and shaping the direction of data gathering;
3. Coalescing of cues into clusters;
4. Activating possible diagnostic explanations (hypotheses);
5. Hypothesis and data-directed search of the data field;
6. Testing for the correct diagnostic hypothesis;
7. Diagnosis.

In general terms, the health care professional meets the patient and starts gathering cues or data, such as signs, symptoms and patient history. Even during this interaction, the health care professional may start to identify the main cues and pieces of data. He then starts to make inferences from this information. This is when short-term and long-term memory come into play, as the health care professional starts to form clusters of data and to recognize patterns. Based on the identified patterns, the health care professional formulates a set of hypotheses that guide the collection of information. It is based on this newly acquired information that the health care professional confirms or refutes a certain hypothesis. It is from the deliberation of these cues and the evaluation of hypotheses that results a diagnosis. In this approach, reasoning is considered to be conscious, speculative and evaluative when the practitioner is solving problems.

An alternative model, still within the analytical framework was suggested by (Carroll and Johnson, 1990). It outlines seven temporal stages of decision-making, namely:

1. Recognition of the situation;
2. Formulation of explanation;
3. Alternative generation of other explanations;
4. Information search to clarify choices and available evidence;
5. Judgement or choice;
6. Action;
7. Feedback.

This approach is more heavily focused on decision-making in the sense that the aim, as opposed to the previous, is to take action and obtain feedback rather than the careful deliberation that characterizes diagnosis. As such, this model is more flexible and fit for situations that require fast decisions (Bryans and McIntosh, 1996; Catriona, 2002).

A common trait to both of the presented decision models is that both use pattern recognition in the process of making a judgement on the basis of a few critical pieces of information. The main characteristic of pattern information is that each new case is compared with previous cases stored in an individual's memory (Offredy, 1998).

The Intuitive Framework

In the intuitive framework, intuition is regarded as an alternative explanation for how health care professionals make decisions. Intuition can be defined as understanding without rationale (Benner and Tanner, 1987), without a clear indicator for something being the case. In the intuitive framework, intuition is crucial for clinical judgement and it is closely linked to the health care professionals' expertise and experience. In Benner (1982), it was possible to observe that the judgement of less experienced professionals was worse than that of more experienced ones. Novice health care professionals rely more on analytical principles to guide their actions, whereas the actions of expert professionals have a component of intuition.

This intuitive aspect of decision-making may be regarded as heuristics, also referred to as rules of thumb. In Buckingham and Adams (2000), it is considered that these rules of thumb are short-cuts created to deal with large amounts of information. For instance, a health care professional may recall the usual pattern or presentation of patients with a particular condition and compare them to the current patient. It is also stated that this intuitive process may occur at an unconscious level, while the analytical process occurs at the conscious level. Therefore, the intuitive explanation lacks the staged structure that the analytical approach shows.

In reality, what is observed is that health care professionals use a combination of the analytical and the intuitive strategies when making a decision, which means that neither offers an exclusive explanation of how decisions are made in a clinical setting (Harbison, 2001; Catriona, 2002). An alternative explanation, provided by Hamm (1988), is the cognitive continuum model. The main principle of this model is that the strategy for clinical decision-making varies on a scale from intuitive to analytical, according to the structure of the task, the time available and the number of information cues. This model is depicted in Figure 1. An example is that, when a task is structured as a experiment (represented as mode 1 in Figure 1), with fewer cues, but more time, the cognitive practice is more analytical and will lead to an analytical judgement. Otherwise, if the task is unstructured, there is little time available, with the potential of lots of cues (represented as mode 6 in Figure 1), then this leads to an intuitive judgement. The intermediary modes represent situations with varying degrees of structure, available time, and number of cues.

The element of memory is undoubtedly more prevalent in the analytical framework. Past cases serve as templates for new cases, and the relationships between symptoms and health conditions that are established, namely in well conducted clinical trials, build the theoretical knowledge that health care professionals must have in order to make decisions that lead to better outcomes. Evidence-based medicine is exactly this, the explicit and judicious use of current best evidence in making decisions about the care of individual

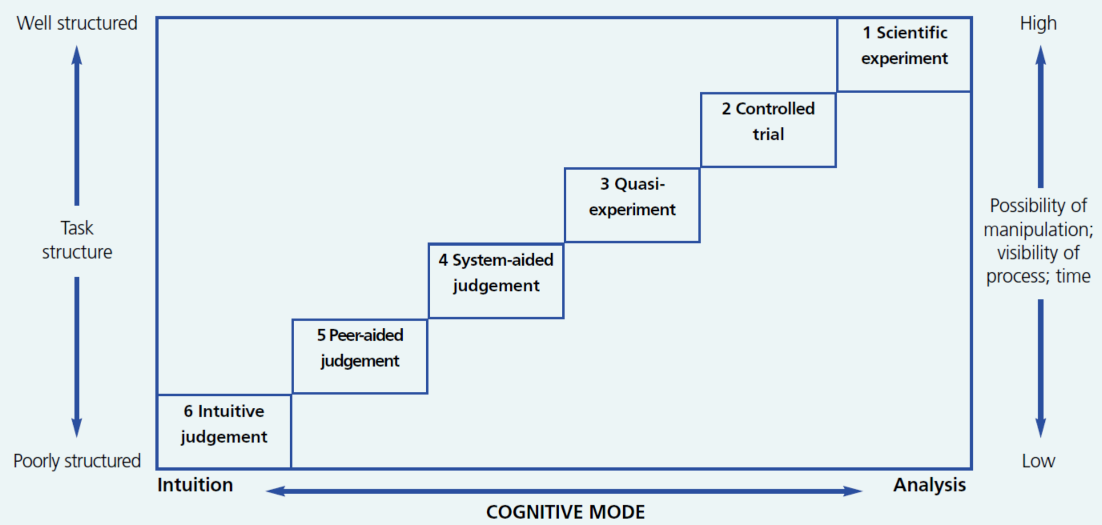


Figure 1: Cognitive continuum model for clinical decision-making (extracted from Muir (2016)).

patients Sackett (1997). It means integrating individual clinical expertise with the best available external clinical evidence from systematic research.

1.1.2 *Uncertainty in Modern Health Care*

As already mentioned, the concept of uncertainty is deeply intertwined with that of decision. Uncertainty is not only an obstacle to the decision-making process, but also one of its defining characteristics.

There are multiple meanings and varieties of uncertainty in health care, which are not often distinguished (Lipshitz and Strauss, 1997; Babrow et al., 1998; Mishel, 1981). According to Han et al. (2011), uncertainty can be defined as the perception of not having knowledge about some aspect of the real world. The form it takes depends on many factors such as the source of uncertainty and how it manifests. In the same work, a comprehensive classification of uncertainty based on the type of issue it raises is provided. The taxonomy diagram is shown in Figure 2 along with examples in cancer treatment. Accordingly, uncertainty can be subdivided into three main categories: scientific, practical, and personal. This division is based on the type of content of the issue. Scientific uncertainty is centred on the disease, whilst practical and personal uncertainties are centred on the health care system and patient, respectively. These three main categories can be further divided into more specific ones. As shown in Figure 2, scientific uncertainty encompasses uncertainties about diagnosis, prognosis, causal explanations, and treatment recommendations. Practical uncertainty applies to the structures and processes of care that the patient is expected to receive from a clinician or institution, or the actions a patient has to undertake in order to receive care. As for personal uncertainty, it is related to psychological and existential issues, reflecting the impact of a disease on the patient's psyche and social life. The non-scientific aspects of uncertainty include issues related with social integration and physician trustability.

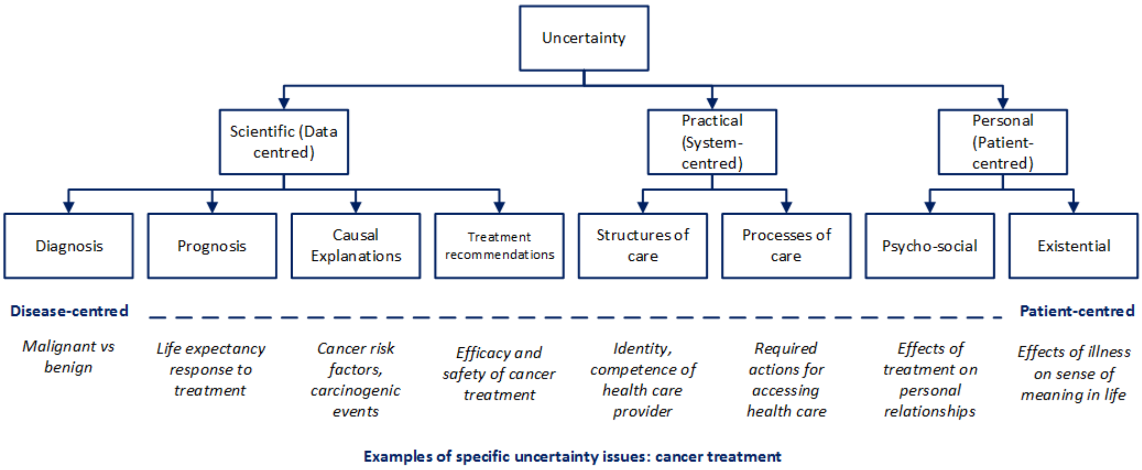

Uncertainty
Scientific (Data centred)
Practical (System-centred)
Personal (Patient-centred)
Diagnosis
Prognosis
Causal Explanations
Treatment recommendations
Structures of care
Processes of care
Psycho-social
Existential
Disease-centred
Patient-centred
Malignant vs benign
Life expectancy response to treatment
Cancer risk factors, carcinogenic events
Efficacy and safety of cancer treatment
Identity, competence of health care provider
Required actions for accessing health care
Effects of treatment on personal relationships
Effects of illness on sense of meaning in life
Examples of specific uncertainty issues: cancer treatment

Figure 2: Uncertainty taxonomy according to the content of its possible issues (adapted from Han et al. (2011)).

Out of the main categories of uncertainty, the one that affects the most the health care professional's decision-making ability is scientific uncertainty, because it is data-centred, and data is the substrate of a decision. The sources of this kind of uncertainty are related with probability, ambiguity and complexity Smithson (1999). Uncertainty derived from probability is related to the indeterminacy of future outcomes and is expressed in terms of numeric probability estimates. Although there are estimates, it is difficult to judge if they are good enough to be the basis of a decision. Ambiguity signifies the lack of reliability, credibility, or adequacy of risk estimates. Complexity captures a source of uncertainty that arises not from the indeterminacy of a phenomenon (probability) or the lack of reliability or credibility about a phenomenon (ambiguity), but from aspects of the phenomenon itself that make it difficult to understand. Another source of scientific uncertainty is incomplete information, discussed in Lipshitz and Strauss (1997), described as the lack of knowledge about the correct values for the parameters of a model. Incomplete or missing information blocks completely the decision-making process, considering that data is vital for the early and intermediary stages, as seen in the description of the analytical process. This later source of uncertainty will be object of study and reflection further ahead in this thesis and, also, one of the issues addressed in the objectives.

1.2 CHALLENGES OF CLINICAL PRACTICE

There is an increasing pressure on health care professionals to standardize their clinical practice in order to prevent undesired outcomes. The trust patients have in their clinicians has a vital importance in clinical interventions, as the lack of it may generate situations of practical uncertainty, related with the structure of care. It is possible to identify three situations that, given their impact in health care, require special attention. They are: medical error, defensive medicine, and undesired variations in clinical practice. These situations may be, themselves, the cause or the consequence of poorly made decisions.

1.2.1 *Medical Error*

Errors are planned activities that fail to achieve their goal, and when such failures are not due to chance alone (Reason, 2000). In a clinical context, a medical error has been considered to be a failure of a planned action to be completed as intended (Kalra, 2004). A concept related to medical error is adverse event, which signifies injuries that result from medical management rather than the underlying disease (Kalra, 2004). Given these definitions, a medical error may or may not result in injury, i.e., it may or may not cause a preventable adverse event.

The causes for the occurrence of medical errors are related with the environment in which decisions take place (Kalra, 2004). Health care is being delivered in an environment with complex interactions among many variables like the disease process itself, the medical staff and equipment, the infrastructure, organizational policies and procedures. And the processes in health care often lack a well-defined structure that allows for a tight control. Some of the key decisions are taken in split seconds, following an intuitive strategy, and are frequently based on little or no prior medical information. As such, it is to be expected that the rates of error are the highest in such challenging environments.

The frequency of adverse events in health care should not be neglected, as this is an issue in health care institutions worldwide. Vincent et al. (2001) showed that, in London hospitals, the rate of adverse events is about 10.8% and, another study in Australian hospitals showed that this number in health care institutions of the country rises to 16.6% (Wilson et al., 1995). According to recently published work by Makary and Daniel (2016), medical error is the third leading cause of death in the United States. European data, mostly from European Union Member States, consistently shows that medical errors and health-care related adverse events occur in 8% to 12% of hospitalizations (World Health Organization, 2016). A survey conducted in the European Union, showed that 26% of patients believe they have suffered an adverse event either directly or indirectly, as shown in Figure 3 (Eurobarometer, 2010).

In Portugal, it is difficult to find statistics about medical error, yet there are cases that emerge and are indicative of how complex the health care environment is and that Portuguese health care institutions are not unaffected. In early 2016, a case in a Portuguese hospital was reported in the news (LUSA, 2016). According to the article, three patients who had surgeries to remove cancerous tumours were not properly directed to the necessary adjuvant therapy, which was, in this case, chemotherapy. The principles of adjuvant therapy for cancer are clearly defined in Clinical Practice Guidelines (CPGs), and chemotherapy, except in special cases, normally follows surgical treatment. This course of action puts the patients' well-being at risk and can have a deep negative impact in their recovery. Although what happened in this case is unclear, it is fair to consider that there was some kind of error in the management of these patients.

Thus, there is a need to provide a better support to health care professionals and make the information they need available at the time and place of care, helping them keep track of their patients and of what they need to do.

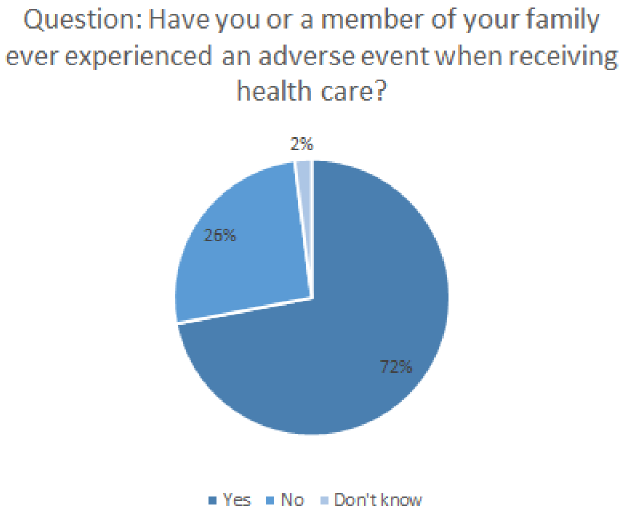


Figure 3: Perception of European Union (while consisting of 27 countries) citizens of having experienced either directly or indirectly an adverse event (extracted from Eurobarometer (2010)).

1.2.2 *Variations in Clinical Practice*

Variations in clinical practice occur during the decision-making process and describe situations in which, presented with the same information, health care professionals take different decisions. They can be divided into two groups: justified variations and unjustified variations (Hampton, 1997).

Justified variations are those that occur due to differences in health care systems, differences in population characteristics (e.g., socio-demographical, cultural, health condition, among others) or due to preferences of the patient or health care professional, in the event that there is no scientifically valid option (Hampton, 1997). This kind of variation stems from the inherent freedom of health care professionals and is desirable, since it allows them to adapt their practice to the conditions of the environment.

Unjustified variations are those that occur when one has the above-mentioned factors controlled, for no apparent reason. They bring no benefit and are harmful for patients, contributing to the deterioration of medical practice. Although difficult to quantify with accuracy, there is evidence that gaps exist between what is known to be effective and based on the best available evidence and research, and what happens in practice. Although there are numerous references reporting this problem, there is little data about variations in clinical practice, as this is a sensible topic and one that is also difficult to study (Wennberg, 1991; Grol, 2001).

The origins of these variations can be pointed out as being occupational stress, lack of accuracy in data collected from patients, and the tendency of health care processionals to follow the practice of the majority, even if it is not supported by the best evidence (Hampton, 1997). This causes difficulties in the assessment of the quality of care, considering there is no standard practice that can be linked to an outcome. Therefore, it is necessary to promote the standardization of health care, while safeguarding the freedom of judgement of health care professionals, as long as it is based on scientific evidence. One way to achieve this is to promote adherence to CPGs, since they are effective means to disseminate the best evidence-based clinical practice (Pearson, 1998).

1.3 CLINICAL DECISION SUPPORT SYSTEMS

Since the emergence of Artificial Intelligence (AI), a term first coined at a famous Darmouth College conference in 1956 by John McArthy (Russell et al., 2003), researchers have been trying to amass its computational power and direct it towards clinical decision-making. This led to the emergence, in the 1970s, of Artificial Intelligence in Medicine (AIM) (Shortliffe, 1993; Patel et al., 2016), which consists in the application of tools and techniques from the field of AI to biomedical problems. One of the tasks that AIM is most concerned with is clinical decision-making. Therefore, CDSSs became one of the flagships of AIM.

According to Musen et al. (2006), a CDSS is *"any computer program designed to help health care professionals to make clinical decisions"*. This definition includes any computer system that deals with clinical data or knowledge in order to provide decision support. According to their function, it is possible to classify CDSSs into:

- **Tools for information management:** specialized knowledge-management workstations for storing, retrieving and browsing clinical information. These tools provide the knowledge and data needed by physicians, but do not help them to apply it. They do not provide any interpretation for the data;
- **Tools for focusing attention:** tools that focus the users' attention to remind them of diagnoses or problems that can be overlooked. Examples include clinical laboratory systems that flag abnormal values;
- **Tools for patient-specific recommendations:** these are programs that are custom-tailored to provide assessments or advice based on sets of patient-specific data. They can follow simple algorithms, symbolic approaches to represent knowledge and reasoning, or models with origins in machine learning.

The separation between these types of CDSSs is not rigid. There are systems that show qualities that place them in more than one category. Nonetheless, this division is useful to showcase the different capabilities in which computer systems can support health care.

1.3.1 *Evolution*

One of the first approaches to CDSSs was the Leeds Abdominal Pain System, from the early 1970s, developed at the University of Leeds (De Dombal et al., 1972). The system used sensitivity, specificity, and disease-prevalence data for various signs, symptoms, and test results to calculate, using Bayes' theorem, the probability of seven possible explanations for acute abdominal pain, namely appendicitis, diverticulitis, perforated ulcer, cholecystitis, small-bowel obstruction, pancreatitis, and nonspecific abdominal pain. The diagnoses of the system were compared with those of real life physicians, and the reports showed that, in 304 cases, the physicians were correct in only 65% to 80%, while the system achieved 91.8% of correct results. However, application in other settings failed to produce better or equal results. This is an example of a system that takes over the decision process and plays the role of a physician, providing also an explanation for the diagnosis and, thus, acting as an Expert System (ES).

The MYCIN system was a consultation system for the management of patients with infectious diseases (Shortliffe et al., 1975). The knowledge of the system consisted of production rules chained together. By removing, altering, or adding rules, the developers could redefine the knowledge base, without explicitly reprogramming or restructuring the other rules. Again, this is an ES that offers explanations about its reasoning and outputs. The differentiating factor from the previous system is that its developers believed straightforward algorithms or statistical approaches were unfit for infectious disease management.

Another early example is the HELP system, developed at the LDS Hospital in Salt Lake City (Gardner et al., 1999). It adds to a conventional medical-record system a monitoring program and a mechanism for storing decision logic in HELP sectors or logic modules. Besides allowing the consultation of information, there was an event-driven mechanism, based on specified criteria about clinical parameters, of specialized warnings, alerts, and reports. This is a classic example of a system that can be placed in the categories of tools for information management and tools for focusing attention. HELP was the first major example of the integration of decision support with other system functions in health care institutions.

These earlier approaches demonstrated that it was possible to encode medical knowledge in machine-interpretable formats. However, they were rarely used by health care professionals and were regarded with scepticism. But, in due time this started to change, given the emergence of personal workstations, the World Wide Web, more intuitive interfaces, and the recognition that technology may play and important role in health care. The growing distress of health care professionals with their workload and the complexity of their tasks, along with the pressure to avoid medical error and improve the quality of care, while saving costs, also contributed to this change of perspective (Musen et al., 2006). At the same time, CDSSs started to position themselves as supporting tools, rather than replacements for health professionals, helping in the formalization ot the decision-making process, the gathering of necessary data for a decision, and the overall management of the clinical process.

This new posture led to the emergence of more recent CDSSs such as ONCOCIN, a rule-based medical expert system for oncology protocol management, developed at Stanford University (Shortliffe, 1986). ONCOCIN was designed to assist physicians with the treatment of cancer patients receiving chemotherapy. It was one of the first systems to model decisions and to sequence actions over time, using a customised flowchart language. Within the system, the history of past events and the duration of actions are the main focus.

Another example is DXplain (Barnett et al., 1987), a CDSS that uses signs, symptoms, and laboratory data to produce a ranked list of diagnoses which might explain (or be associated with) the clinical manifestations. DXplain provides justification for why each of these diseases might be considered, suggests what further clinical information would be useful to collect for each disease, and lists what clinical manifestations will ensue. The key word here is suggest, which reflects the new framing of CDSSs within decision support.

Now, well into the twenty first century, with the advent of machine learning techniques, deep learning, and big data, new and sophisticated AIs are being developed . The most impressive examples are arguably IBM's Watson and Google DeepMind's AlphaGo, both with extraordinary feats under their belt. Efforts are underway to adapt these AIs to clinical decision support (Murdoch and Detsky, 2013), particularly in Watson's case. Recently, it was reported by the University of Tokyo that IMB Watson spotted a 60 year-old woman's rare form of leukaemia, a diagnosis that had eluded her doctors for months (News Health Care IT, 2016). The AI compared the patient's genetic changes with a database of 20 million cancer research papers. The supercomputer's diagnosis led to the right treatment for the patient.

1.3.2 *Impact*

When looking at systems such as DXplain (Barnett et al., 1987), it is clear what they do. They emulate the decision-making process of a health care professional in its most analytical form. One of the stages of this process at which CDSSs are more helpful is the gathering of information. Given the growing amounts of data and knowledge based on which health care professionals have to make decisions, CDSSs offer platforms for the management of this volume of information.

Depending on how their knowledge base is organized, CDSSs help define the complex interactions between the variables in the health care environment. They organize information in ways that facilitate decision making. Additionally, CDSSs support the formulation of hypotheses, ensuring that all possibilities are covered. They may also analyse hypotheses and rank them by a metric derived from their decision model (Musen et al., 2006).

CDSSs have been shown to influence physician behaviour by promoting the adherence to the best clinical practice and the avoidance of undesired variations (Schedlbauer et al., 2009; Scheepers-Hoeks et al., 2013). They also have a positive impact on patient safety (Bates and Gawande, 2003) and provide a rationale for diagnostic test ordering and other care processes (Bright et al., 2012), thus reducing the occurrence of medical errors and health care costs. For this, it can be considered that they offer solutions to meet the current challenges of clinical practice. However, the evidence supporting the impact of CDSSs on the outcomes of care are scarce, and there are few studies that corroborate it.

1.3.3 *Guideline-based Clinical Decision Support Systems*

A wide variety of techniques have been used in the design and implementation of decision models in CDSSs. They range from simple logics, such as specifically designed flowcharts for particular problems, to methods drawn from Bayesian modelling, decision analysis with utilities associated to outcomes, or artificial neural networks (Sim et al., 2001). However, these models are highly tailored to the type of problem their corresponding systems address. In order to increase the reach of a system (if that is actually the objective), its underlying decision model and knowledge base have to cover a wide range of clinical moments, namely multiple situations of diagnosis and management, across different clinical categories and specialities. An ideal support for such CDSSs are CPGs.

Characteristics

Guideline-based CDSSs have CPGs as their underlying support. These documents are systematically developed statements to assist health care professionals and patients about appropriate health care in specific clinical circumstances (Miller and Kearney, 2004). CPGs are developed for a wide range of clinical situations and contain advice based on the best available evidence. They are often called other names such as clinical protocols or practice policies. The potential benefits of their inclusion in CDSSs are the reduction of undesired variations in clinical practice, the improvement of outcomes, reduction of medical error, and cost containment (Woolf et al., 1999). CPGs typically represent a medical expert consensus regarding the screening, diagnosis, or management, over either limited or extended periods of time, of patients who have a particular clinical problem, need, or condition.

CIGs are the machine-readable format of CPGs for automatic interpretation in CDSSs (Ten Teije et al., 2008; Peleg, 2013). The interaction with systems that integrate these machine-interpretable formats follows the structure of a dialogue between the care provider and the system. Since CPGs, and thus CIGs, are similar to algorithms with stages connected to each other and receiving information from each other, they resemble the stages of the analytical framework. CIGs define the information necessary for making a decision, elicit options for that decision, and drive the clinical process to the next step.

The development of CIGs started around the late 1980s. Since then, significant approaches to CIG decision support emerged, namely Arden Syntax (Hripcsak, 1994), the Guideline Interchange Format (GLIF3) (Boxwala et al., 2004), PROforma (Fox et al., 1998), Asbru (Shahar et al., 1998), the GuideLine Acquisition, Representation and Execution (GLARE) (Bottrighi et al., 2006), and the Standards-based Sharable Active Guideline Environment (SAGE) (Tu et al., 2007), possessing their own knowledge acquisition and execution tools.

Limitations

The development of CIG-based clinical decision support systems thrived throughout the 1990s and early 2000s (de Clercq et al., 2004), yielding models and supporting systems that focus on different aspects of clinical practice. The result from these fragmented approaches is that there is no standard approach to CIG modelling and execution. These CIG approaches show limitations at several levels that affect, to an extent, the adoption of CIG-based CDSSs:

- Starting with their underlying models, they typically focus on one single aspect of CIGs, paying less attention than the required to the remaining aspects. Such is the case with Arden Syntax (Hripcsak, 1994) and its rather simplistic approach to the modelling of clinical recommendations, possessing knowledge primitives for only one type of situation in the clinical process, a decision, without the possibility to chain recommendations together. Or even GLIF3 (Boxwala et al., 2004), which is severely limited in terms of the representation of temporal constraints;

- Another issue is that these CIG models require the user to have proficiency in programming languages for the definition of constraints regarding clinical recommendations. These languages are used to express which variations in the clinical process trigger sets of recommendations. An example of this is the use of the GELLO (Sordo et al., 2003) language, developed for GLIF3. This may be a deterrent to users who do not have such proficiency from adopting a CIG system, considering that their learning curve would be long;

- The complexity of the models, with numerous proprietary specifications, is also a factor to take into account, PROforma (Fox et al., 1998) and Asbru (Shahar et al., 1998) are paradigmatic cases in this regard. The knowledge acquisition systems for CIGs, whose purpose is to enable and facilitate the encoding of CPGs in machine-interpretable formats, lack features that reduce this complexity (Isern and Moreno, 2008);

- The support to the clinical decision-making process has also been neglected. The current CIG execution systems do not provide complements to the CIG knowledge base and the decision is made with the simple matching of clinical parameters with the premisses of CIG rules (Isern and Moreno, 2008). Although CIGs are machine-interpretable versions of CPGs, they still carry some of the criticism that is directed to their textual counterparts, namely that they are strict practices that are difficult to adapt to the context. There are situations that CIG systems cannot accommodate, one of them being the problem of incomplete information in data entry points. CIG execution is driven by data, and, if it is missing, the system is unable to proceed with the decision-making process and produce a recommendation. There are also tasks, such as prediction (e.g., of the evolution of patients) and classification (e.g., categorization of patients according to signs and symptoms), that CIG execution systems either cannot handle or they manage based on static and stiff models that barely reflect the context of the patients;

- The way in which CIG execution systems provide advice should also be object of reflection. The structured dialogue that these systems typically follow may be unfit for every moment of clinical practice. There is a need of methods to integrate and increase the adoption of CPGs through CIG-based CDSSs in order to prevent cases such as the medical error reported in Section 1.2.1;

Addressing these limitations is the underlying motivation for this doctoral thesis and the driving factor of all the work that supports it.

1.4 RESEARCH HYPOTHESIS

Taking into account the challenges that CIG-based CDSSs face, which were pointed out in Section 1.3.3, there should be solutions that address them. The objective of any computational system should be to enhance the functions of the rudimentary systems that were in place before them. However, this is clearly not he case when it comes to the limitations of CIG-based CDSSs. These very same limitations can be pointed out to the paper versions of CPGs, which means that they were transported to their computational implementation. This is particularly true in the complexity of the models, since CPGs are, themselves, complex documents that are difficult to interpret. Yet, their implementation in systems for automatic reasoning did not smooth out this complexity. Additionally, the style of communication of these CIG-based CDSSs and the lack of higher-level decision-making functions (they are purely analytical without any component that could resemble some kind of intuition) makes the interaction with them little different than reading their document counterparts.

The point here is that there is a need to approximate CPGs to the concept of *living guidelines* (Seyfang et al., 2007), which conveys desirable properties for these documents such as modularity and interactivity. These properties translate into CPGs that are easier to create and modify, that have an increasing presence in a clinical setting, and are more flexible and adaptable. It is considered that the use of CIGs in CDSSs is a step forward in that direction, but, as seen, the idea is still far from being completely fulfilled.

The goal of this thesis is, thus, to answer these challenges by turning to AI techniques such as ontology specification, logic programming and machine learning, and to web technologies to outline a framework for the expression of CIG-based CDSSs. Of course that there is also a social motivation underlying this work. By increasing the intervention of CPGs in a clinical setting, through the improvement of CIG-based CDSSs, one is also promoting the adherence to best practices and, therefore, potentially reducing medical error and undesired variations in clinical practice.

To conclude, the **research hypothesis** for this work is formulated as follows:

The proposed framework for CIG-based CDSSs enables the expression of properties of living guidelines, namely modularity, flexibility, adaptability, and interactivity.

The same is to say that, through the application of the studied AI techniques and technologies, it is possible to answer the current challenges of CIG-based CDSSs.

1.5 OBJECTIVES

The title of this doctoral thesis is "Clinical Decision Support: Knowledge Representation and Uncertainty Management". As such, the developed work focuses on the matters of knowledge representation in CDSSs, reasoning with this knowledge, and the underlying decision-making process. Given the exposition about CDSSs, the limitations of CIG systems, and the research hypothesis formulated for this thesis, the main goal of the work is to conceive a framework for the expression of CIG-based CDSSs that contributes to increase their flexibility and adaptability, so that they can become more modular and interactive. The term flexibility is used here to convey the ease with which the knowledge in the system is moulded to accommodate new domains. As for adaptability, it refers to the capacity of the execution system to adapt to situations of uncertainty, namely that of scientific uncertainty.

This main goal was approached in terms of the following objectives:

1. Identification and characterization of the main requirements and necessities of CIG systems;
2. Identification and analysis of techniques and models stemming from AI and web technologies that can provide the capacities to CIG systems to overcome their challenges.
3. Formalization of a comprehensive CIG representation, fit for multiple medical domains and situations in the clinical process;
4. Development of a library of guideline items with re-usable knowledge patterns that facilitates the encoding of CPGs in a CIG format;
5. Definition of an architecture for the agile deployment of a CIG-based CDSS;
6. Suggestion of a method to achieve a higher integration of CIG recommendations in clinical practice;
7. Development of a mechanism to manage data-centred uncertainty, namely incomplete information, in the decision points of CIGs;
8. Exploration of dynamic models, for prediction and classification, that offer a complement to CIG recommendations.

Following the enumeration of relevant aspects of CIG-based systems proposed by de Clercq et al. (2004), namely guideline acquisition, guideline verification and testing, and guideline execution, this work focuses on the aspects of acquisition and execution, with the addition of decision-making, from the point of view of the availability of the information.

1.6 RESEARCH METHODOLOGY

In order to develop this doctoral thesis, the Action Research methodology (O'Brien, 1998) was employed. It is a holistic approach to problem-solving, based on continuous development, and directed to problem-solving. What distinguishes this approach from normal, everyday problem-solving is the emphasis it puts on scientific study. The researcher systematically studies the problem and his interventions are backed by theoretical considerations.

The process starts with a research hypothesis, based on a thorough review of the state of the art, in this case of CIG-based CDSSs. From this step results a report of available resources and features. In the context of this thesis, and faced with what was exposed in Section 1.3.3, the hypothesis is defined in Section 1.4. After the formulation of this hypothesis, a set of objectives is established. They pave the way to proving or refuting the research hypothesis. Accordingly, the objectives for the thesis are outlined in Section 1.5. Since they can be grouped into three different aspects of CIGs, namely CIG acquisition, CIG execution, and CIG decision-making, the set of steps defined by Action Research were applied to each one of these aspects separately. There are, in total, six steps modelled in Action Research. They are the following:

1. **Definition of the problem and issues:** this step states the problem and investigates what originates it, gathering all its features and formulating a hypothesis;
2. **Constant update of the state of the art:** the review of the state of the art studies the features of all the projects related to the current research. The update of this component allows for the adjustment of objectives and the pursuit of new solutions;
3. **Design of a solution:** the information gathered in the previous steps enables the design of a solution that fulfils the specified objectives, which, in turn, will allow to prove or refute the research hypothesis;
4. **Implementation and experimentation through prototypes:** formalization of a prototype that contains all the features specified in the solution. Its behaviour is then observed in order to ascertain its efficacy;
5. **Analysis, validation and conclusion:** this step consists in the analysis and validation of the prototype. This is done by checking if the implementation achieves the objectives, which leads to conclusions regarding the research hypothesis;
6. **Dissemination of results in the scientific community**: dissemination of research results in peer-reviewed journals, conferences, workshops, among others.

Based on this last step of Action Research and the logical sequence of publications that resulted from the developed work, it was decided that the presentation of this thesis would follow the scheme of publication compilation.

1.7 DOCUMENT GUIDE

This section aims to be a guide for helping readers to easily find the right content in this thesis and, also, to understand some aspects related with the structure of the document.

This thesis is presented under the scheme of publication compilation and, therefore, some requirements have to be met in order to make it understandable and coherent. The first content to be provided is a summary of the presented doctoral thesis, both in English and Portuguese. This is followed by the necessary Table of Contents, List of Figures, List of Tables, and Acronyms.

The thesis can be divided into three main blocks: Introduction (Chapter 1), Publications Composing the Doctoral Thesis (Chapter 2), and Conclusions (Chapter 3). At the end of the thesis, the Bibliography used in the Introduction and Conclusions is presented.

Each chapter of the thesis has a well-defined purpose. As such, a brief summary of each one is presented, with more focus on Chapter 2, which contains the compilation of publications. The description is as follows.

Chapter 1: Introduction

This chapter introduces the thesis with the exposition of concepts that are relevant to the understanding of its scope and purpose. The topics covered by the chapter include clinical decision-making, uncertainty in health care, the current challenges of clinical practice, and CDSSs. On this last topic, a special attention is given to the characterization of CIG-based CDSSs and their limitations.

Based on these limitations and the challenges they present, the research hypothesis and the objectives for the thesis are defined. Following this, the research methodology is outlined. The section ends with a document guide describing the structure and contents of the thesis.

Chapter 2: Publications Composing the Doctoral Thesis

This section contains the publications selected to integrate the doctoral thesis. It consists of six sections, each one representing a different scientific publication. At the beginning of each section there is a table with relevant information about the featured publication.

Section 2.1: Development and Implementation of Clinical Guidelines: An Artificial Intelligence Perspective

The publication featured in this section characterizes CPGs and CIGs, providing definitions and examples. Six CIG approaches, namely Arden Syntax, GLIF3, PROforma, Asbru, GLARE and SAGE, are described in terms of their strengths and limitations in the acquisition and execution of CPGs. The section also covers ESs, case-based reasoning (CBR), medical ontologies, and logic programming as solutions to meet the requirements of a CIG-based CDSS. It is a review of the state of the art in the field.

Section 2.2: Representation of Clinical Practice Guideline Components in OWL

This section presents the CompGuide ontology for the representation of CPGs. The publication featured in the section describes the fundamentals about Web Ontology Language (OWL), and the advantages of choosing this knowledge formalism. The representation primitives of the ontology are presented with a brief example. The publication ends with an analysis of the advantages of the model in comparison with existing ones.

Section 2.3: Assessing an Ontology for the Representation of Clinical Protocols in Decision Support Systems

The publication included in this section describes an experiment conducted in order to assess the expressiveness of the CompGuide ontology. In it, 14 students were asked to represent a set of CPGs according to the ontology. They were then asked to evaluate the ontology through a questionnaire regarding the expressiveness of the model for different types of content, particularly in the representation of administrative information, the construction of workflow of procedures, the definition of temporal constraints, and the definition of clinical constraints. The answers were provided in a five point Likert scale demonstrating the agreement with sets of statements. The publication ends with a critical analysis of the results.

Section 2.4: Decision Support Provided by a Temporally Oriented Health Care Assistant

The publication occupying this section reports the development of a health care assistant tool for the execution of CIGs. It provides a description of the CompGuide temporal model, which is crucial for the development of the assistant. There is also a comparison of this model with other existing temporal models. Additionally, there is a description of the CompGuide system which supports all the logics behind the execution of CIGs against patient data. To demonstrate the full expressiveness of the temporal model, a case study with a guideline for colon cancer treatment is presented. To end the section, the temporally oriented functionalities of the health care assistant are described.

Section 2.5: A Dynamic Default Revision Mechanism for Speculative Computation

This section is dedicated to the explanation of Speculative Computation, a theory from logic programming for the management of incomplete information in decision-making steps, based on default constraints. The theory features a dynamic revision of default constraints supported by Bayesian networks built from past data about the decision problem in question. The publication provides related work in the fields of constraint logic programming and defeasible reasoning. A CIG-based CDSS is used as a case-study to showcase the speculative mechanisms and their effects.

Section 2.6: A Mobile and Evolving Tool to Predict Colorectal Cancer Survivability

The publication featured in this section proposes a tool for survivability prediction of patients with colorectal cancer. It provides a state of the art in the field and identifies limitations in existing approaches, defining the requirements that a tool performing such a task should have. The architecture of the whole prediction system is described as being service-oriented. The prediction models and an interface example are showcased. This system performs a complementary function to CIG-based CDSSs.

Chapter 3: Conclusions

In the Conclusions there is a description of the contributions resulting from the thesis and of how they validate the research hypothesis. In addition, the activities undertaken for the dissemination of results are enumerated. The chapter ends with final remarks and considerations about future work.

2

PUBLICATIONS COMPOSING THE DOCTORAL THESIS

2.1 DEVELOPMENT AND IMPLEMENTATION OF CLINICAL GUIDELINES: AN ARTIFICIAL INTELLIGENCE PERSPECTIVE

Title	Development and implementation of clinical guidelines: An artificial intelligence perspective
Authors	Tiago Oliveira, Paulo Novais, and José Neves
Publication Type	Journal
Publication Name	Artificial Intelligence Review
Publisher	Springer Netherlands
Volume	42
Number	4
Pages	999–1027
Year	2014
Month	December
Online ISSN	1573-7462
Print ISSN	0269-2821
URL	http://link.springer.com/article/10.1007/s10462-013-9402-2
State	Published
Scimago journal rank (2014)	1.013, Artificial Intelligence (Q2), Language and Linguistics (Q1)
JCR impact factor (2014)	2.111, Computer Science (Q2), Artificial Intelligence (Q2)

Contribution of the doctoral candidate
The doctoral candidate, Tiago José Martins Oliveira, declares to be the main author and the major contributor of the paper *Development and implementation of clinical guidelines: An artificial intelligence perspective*.

Artif Intell Rev
DOI 10.1007/s10462-013-9402-2

Development and implementation of clinical guidelines: An artificial intelligence perspective

Tiago Oliveira · Paulo Novais · José Neves

Abstract Clinical practice guidelines in paper format are still the preferred form of delivery of medical knowledge and recommendations to healthcare professionals. Their current support and development process have well identified limitations to which the healthcare community has been continuously searching solutions. Artificial intelligence may create the conditions and provide the tools to address many, if not all, of these limitations.. This paper presents a comprehensive and up to date review of computer-interpretable guideline approaches, namely Arden Syntax, GLIF, PROforma, Asbru, GLARE and SAGE. It also provides an assessment of how well these approaches respond to the challenges posed by paper-based guidelines and addresses topics of Artificial intelligence that could provide a solution to the shortcomings of clinical guidelines. Among the topics addressed by this paper are expert systems, case-based reasoning, medical ontologies and reasoning under uncertainty, with a special focus on methodologies for assessing quality of information when managing incomplete information. Finally, an analysis is made of the fundamental requirements of a guideline model and the importance that standard terminologies and models for clinical data have in the semantic and syntactic interoperability between a guideline execution engine and the software tools used in clinical settings. It is also proposed a line of research that includes the development of an ontology for clinical practice guidelines and a decision model for a guideline-based expert system that manages non-compliance with clinical guidelines and uncertainty.

Keywords Computer-interpretable guidelines · Ontologies · Decision support · Quality of information

T. Oliveira (✉) · P. Novais · J. Neves
CCTC/DI, Department of Informatics, University of Minho,
Campus of Gualtar, 4710-057 Braga, Portugal
e-mail: toliveira@di.uminho.pt

P. Novais
e-mail: pjon@di.uminho.pt

J. Neves
e-mail: jneves@di.uminho.pt

Published online: 13 March 2013

1 Introduction

There is an increasing pressure in healthcare professionals to standardize their clinical practice in order to prevent undesired variations. Clinical practice guidelines (CPGs) are developed in order to achieve this purpose. In recent years there has been an explosion of interest in CPGs, with initiatives to stimulate guideline development promoted by many countries and healthcare institutions. In fact, CPGs are, currently, the best way to convey information to healthcare professionals, to ensure that their clinical practice follows the rules of medical procedures. This is a very important matter, if one takes into account the consequences that may arise from a poorly conducted clinical process. The prevalence of medical errors is significant in hospitals across the world (Brennan 2000; Kalra 2004). Putting aside the human cost, which is immeasurable, the economic cost from lawsuits and other legal issues resulting from medical error has a deep impact in the budget of healthcare institutions. However, an overzealous practice, like defensive medicine, may have equally undesired consequences (Chawla and Gunderman 2008). The prescription of exams and treatments without scientific proof or basis also has a great economic impact and may seriously undermine the confidence that patients have in their physicians. This has consequences in the mental health of patients as well. The primary objective of CPGs is to provide a scientific support to clinical procedures, thus mitigating the occurrence of these situations.

However, healthcare professionals still show some resistance towards complying with CPGs. The arguments used to justify this behavior are that guidelines stifle change and innovation, and restrain clinical practice, preventing healthcare professionals from adapting their practice to their social, economic and cultural contexts (Thomson et al. 1995). Guidelines evolved in order to address some of this criticism, through the development of mechanisms to smooth updating processes and to accommodate justified variations in clinical practice. Currently, we live in the age of information and, once again, CPGs should evolve to keep up with the rapid growth of scientific knowledge. Research in the field of computer-interpretable guidelines (CIGs) is booming, due to the need to deliver information to healthcare professionals in a faster way and to support them in decision making.

This paper starts by providing some background information on CPGs and how they are developed. Then some artificial intelligence (AI) techniques will be addressed with the objective of determining how AI can improve the current state of the art as well as the development and the execution of CPGs. The final sections of the paper provide a general presentation of the research line that is being followed and how it integrates the conclusions extracted from the analysis made. It goes without saying that a paper format cannot be compared to a computerized guideline, as the first cannot be processed electronically. The perspective this work intends to show is how a digital format can be more advantageous and provide a new set of tools to facilitate the work of healthcare professionals.

2 Clinical practice guidelines

2.1 What are clinical practice guidelines?

CPGs are systematically developed statements to assist healthcare professionals and patients about appropriate healthcare in specific clinical circumstances (Miller and Kearney 2004). This is the most widely accepted definition of clinical guideline, provided by the Institute of Medicine, of the United States (US). There are other terms used as synonyms of CPGs such as protocols, practice policies, clinical policies, practice parameters and clinical pathways.

Usually, the name given to these documents is a matter of personal preference rather than a reference to a standard nomenclature and it can change across healthcare institutions and countries. Despite these differing nomenclatures, there are common objectives associated (Miller and Kearney 2004) with all of them, such as:

- Help healthcare professionals and patients in decisions about clinical procedures;
- Describe appropriate care based on scientific evidence;
- Act as the focus for quality assessment and activity improvement, including audits.

CPGs are decision tools devised to shorten the distance between real clinical practice and optimal clinical practice (Mead 2000). The potential benefits from the implementation of CPGs include the reduction of morbidity and mortality, efficiency improvement and cost containment. They also provide their users with a reference by which they guide their clinical practice, and measurable criteria to assess their performance. The evidence contained in CPGs is used, at the same time, to inform healthcare professionals of the latest developments in scientific knowledge and to justify their decisions during the clinical process (Thomson 2000).

The format of these documents is not standardized and shows variations according to the organization producing the guideline and the clinical area it addresses. Since the middle of 1990s, many worldwide organizations started evidence-based CPG development programs, namely the Scottish Intercollegiate Guidelines Network (SIGN), the New Zealand Guidelines Group (NZGG), the Guidelines Advisory Committee (GAC) in Canada, and the National Federation of Cancer Centers (FNCLCC) in France, among others (Rosenbrand et al. 2008). These organizations joined others that paved the way for guideline development like the Institute of Medicine and the Dutch Institute for Healthcare Improvement. In 2002, an international effort towards the dissemination of CPGs culminated in the creation of the Guidelines International Network (G-I-N). (Ollenschläger et al. 2004). Currently, this global network comprises 92 organizations and 127 individual members, representing 48 countries, putting forth efforts in order to standardize guideline development and implementation. In recent years, some online guideline repositories started to appear, among which should be highlighted the National Guideline Clearinghouse (NGC) of the US. NGC[1] is a public resource for evidence-based CPGs and gathers guidelines from various organizations under different labels that represent their category and medical specialty.

2.2 Development of clinical practice guidelines

Each organization follows its own guideline development process. However, the different development methodologies have common phases and follow similar principles.

Initially, guidelines were only based on the consensus of groups of experts, but with the growth of evidence-based clinical practice, other techniques were included in guideline development. The Delphi and nominal group techniques are some of the methodologies that were later included in the development process and are still used today (Hutchings et al. 2006). Currently, guideline development is more focused on an extensive research of the literature and thorough analysis of empirical evidence.

The process usually starts with the choice of the guideline topic or subject, based on the problems that motivate the development (Rosenbrand et al. 2008; Rosenfeld and Shiffman 2006). CPGs can be developed to a wide range of subjects and medical areas, including health

[1] http://guideline.gov/index.aspx.

conditions bound to diseases and economical costs. To choose the topic, it is necessary to do a preliminary check of the available evidence in order to ascertain the validity of the theme.

The composition of the work group is the following step (Rosenbrand et al. 2008; Rosenfeld and Shiffman 2006). The efficiency of the guideline highly depends of the nature of the group producing it. The work group must be multidisciplinary, in a way that includes participants from all the areas affected by the topic of the guideline. Once the group is gathered, the analysis of the underlying problem, to which the guideline must provide a solution, starts. The work group must search for other guidelines concerning the topic, whose existence does not invalidate the creation of a new one as the existing ones may be outdated. The result from the analysis of the problem should be a set of key-questions that clearly identify the population being studied (the group of individuals who will be the target of the diagnosis or intervention), the type of control used and the efficiency measures that will be used to evaluate the interventions.

The objective of the literature research is to find the best available evidence, capable of answering the key-questions formulated in the previous step. The development group has to define some search constraints (e.g., to privilege a published work over an unpublished one) in order to assure the quality of the evidence. Once all the information sources are gathered, the work group does a critical appraisal of the evidence, based on the methodologies used to do the studies that generated them. The reviews are summarized in evidence tables, with a grade being given to the medical trials that were selected (Rosenbrand et al. 2008; Rosenfeld and Shiffman 2006). Healthcare institutions that produce guidelines do not have a common grading system, which is inconvenient when one has to compare the evidence of similar guidelines. The Grading of Recommendations Assessment, Development and Evaluation (GRADE) (Kavanagh 2009) workgroup was created with the objective of developing an approach to evidence grading that can be used by different organizations. The GRADE system has been adopted by an increasing number of organizations and it is in continuous development.

After the evidence grading, the workgroup must elaborate a sketch of the guideline and submit it to external revision. Usually, this revision is performed by independent entities in conferences or healthcare-related gatherings. This is an iterative process in which the guideline is altered according to the reviews and then proposed for another external revision, until it reaches a stable version. Then the guideline is published and disseminated through conferences and newsletters to healthcare professionals.

2.3 Shortcomings of clinical practice guidelines in the present

In the final phase of the development process, the development group has to choose suitable means for disseminating guidelines (Thomson et al. 1995). The usual ones are newsletters to healthcare professionals, disclosure at medical conferences and through online PDF repositories of guidelines (Cheater and Closs 1997; Dennis et al. 2004). However, these means do not provide the desired coverage and sometimes fail in the delivery of knowledge to healthcare professionals. This is an important aspect because feedback from the medical community is the best mechanism through which guidelines are improved.

Guideline documents have a structure that makes them difficult to consult. Usually they are long texts and the clinical recommendations are contained in the body of that text. This aspect interferes with the retrieval of relevant information by healthcare professionals and makes the consultation for real time application rather complicated. Moreover, these long documents are difficult to update, which is a great drawback in the evolution of a guideline.

They should accompany the development of clinical knowledge in a specific medical area (Rosenbrand et al. 2008).

Another issue is the ambiguity of the content of guidelines (Woolf et al. 1999). Ambiguity can be classified into syntactic, semantic and pragmatic (Codish and Shiffman 2005). Syntactic ambiguity occurs when the structure of a statement is not clear, thus impeding its correct interpretation. Misplaced (or lack of) punctuation and wrongfully applied Boolean connectors are some of the causes of syntactic ambiguity. The classic definition understood generically by people as ambiguity fits the category of semantic ambiguity, characterized by situations in which terms can be interpreted in more than one way. Misuse of abbreviations, such as the case of the word "cold", which in the context of a guideline can mean "common cold", "cold sensation" or "Chronic Obstructive Lung Disease", fall in the spectre of semantic ambiguity. As for pragmatic ambiguity, it happens when the recommendations of CPGs are not consistent or are conflicting with each other.

The vocabulary used in CPGs may also denote vagueness (Codish and Shiffman 2005). Sometimes the boundaries of a term are not completely understood by healthcare professionals. To show an example, temporal vagueness is frequent in guideline recommendations, with the use of terms such as "rare" or "common". The poor specification of terms is also frequent, with terms like "moderate", "elderly" and "adequate" being used without sufficient detail for clear interpretation. The texts often have occurrences of probabilistic terms to describe the frequency of events, namely "impossible", "certain", "unlikely" and "probable", whose interpretation falls upon the subjective perception of the reader. The same situation occurs with some of the quantitative terms that are used.

Healthcare professionals often complain that, rather than offering support for clinical practice, CPGs restrain it, the argument being that they do not consider the social, cultural and economic conditions of the context in which they are applied (Woolf et al. 1999). Healthcare professionals may need to adapt their clinical practice according to the origin of their patients, but the steps for doing so are not described in CPGs. This lack of context-awareness is one of the major causes of noncompliance.

Currently, CPGs do not cope with preference-sensitive decisions, for instance, between scientifically valid treatments that may be applied to the same situation. In this case, there should be a group decision that takes into consideration the preferences and goals of the medical team responsible for the clinical case as well as those of the patient (Weijden et al. 2011). What usually happens, in these cases, is that the decision is made by one healthcare professional only, without consulting the other parts involved.

The level of uncertainty and incompleteness of the information upon which decisions are made, during the application of guidelines, is also a matter of concern (Logan and Scott 1996). A symptom is a somewhat uncertain indication of a health condition as it may or may not occur together with the disease. Thus, it is necessary a measure of the uncertainty associated to the observation of a symptom and the risk of the occurrence of a disease. During clinical encounters, healthcare professionals have to collect the values of relevant clinical parameters that build the patient's health state. The observations made by healthcare professionals in order to obtain these values have a subjective nature, mainly because a human being is doing them, thus the information they generate may be contradictory/inaccurate and sometimes the values of these parameters may not be obtainable due to the lack of technical means to do so. These cases of contradictory, inaccurate and missing information fall under the designation of incomplete information.

3 Artificial intelligence and clinical guidelines

AI is a field of study that aims to explain and emulate intelligent behaviour in computational processes (Schalkoff 1990). It is the branch of computer science that is concerned with the automation of intelligence. The ability to make machines think like human beings creates new possibilities in many areas. Research in AI helped the development of new technologies that nowadays are the basis of many big systems. These technologies are primarily used to automate tasks and improve knowledge-based processes, such as decision making.

The application of AI in medicine can be traced back to the middle of the 1970s and early 1980s, and led to the appearance of a subarea in AI, called artificial intelligence in medicine (AIM). Research in this new field was pioneered by research groups in the US. An early definition of AIM was provided by Shortliffe (1993), stating that the primary concern of this research area was the construction of AI programs that perform diagnosis and make therapy recommendations. This definition reflects the primary focus of AIM at that time, which was the understanding and automation of the clinical encounter. Nowadays, AIM is more focused on giving support to healthcare workers rather than trying to replace them. As so the identification of the right areas of medicine in which this support can be given is the key aspect that dictates the acceptance of AI technologies by clinicians.

The variety of roles AI programs may play in medicine is very wide. The use of medical knowledge is one of such roles, namely the support to human cognition that can be implemented, for instance, as reminder systems that alert healthcare professionals of clinical events or contradictions in treatment plans. AI programs can also be used to create new knowledge by discovering new phenomena through data analysis, pattern discovery and associations. Machine learning is the subfield of AI that deals with the generation of new knowledge and includes different techniques to produce systems capable of providing a description of clinical features. Case-based reasoning (CBR) is one of such techniques. Based on past clinical cases, CBR is able to generate recommendations to new ones. The form in which these recommendations are provided varies, but rules and decision trees are among the most commonly used. An example of this type of system is KARDIO, for interpreting ECGs (Bratko et al. 1989). Another application of machine learning in medicine is the use of datamining in the construction of pathophysiological models and drug discovery. AI systems containing medical knowledge, usually about a specific domain, are capable of reasoning and reaching conclusions based on data. The array of functions AI programs can perform includes: alerts and reminders, diagnostic support, agents for information retrieval and image recognition/interpretation. DXplain (Barnett et al. 1992) and HELP (Gardner et al. 1999) are examples of these knowledge-based decision support systems and are among the first ones to be developed.

In the remainder of this section, the focus will be placed on some topics and technologies of AI that may provide effective responses to the shortcomings of CPGs and help the development of clinical practice.

3.1 Group decision making

Group decision is a common phenomenon in human decision making activities. It is an arduous task because it implies the aggregation of individual alternatives to yield a decision that is acceptable to the group as a whole (John et al. 2008). The group explores a number of alternative solutions, answering *what-if* questions and the participants may have different roles in the decision process, according to pre-established criteria by the organization.

During the application of CPGs, there are moments when this type of decision is required. The selection among scientifically valid options during the clinical process must be done based on the opinions of the parts involved (healthcare professionals and patients). Technology-assisted decision making may help the generation of ideas and actions, the choice of alternatives and the negotiation of solutions. The existence of CIG models and a tool for execution of CPGs enables the implementation of automated group decision making.

The work of Karacapilidis and Pappis (1997) summarizes some of the aspects that must be taken into account when developing a framework for group decision. The first one is the spatial distance between decision makers and the electronic communication facilities that enable them to communicate with each other. In the clinical setting, it is not uncommon for a clinical case to be treated by a medical team whose members are from different healthcare institutions, so the development of a virtual environment that enables the communication between them may be an advantage to the discussion of guideline recommendations, as shown in previous works with successful knowledge exchanges (Anogianakis et al. 1998; Househ et al. 2011). The type of environment influences the goals of decision makers. The goals are different in an environment where the group wants to solve a common problem cooperatively from another in which bargaining takes place. Typically, in a clinical environment both situations can occur, a medical team may be discussing the diagnosis of a patient and their members may have different opinions based on different evidence. The implementation of techniques in a virtual clinical decision environment to extract information about the actors (e.g., stress level) (Novais et al. 2012), would assist the definition of the type of interaction between the group members and consequently the selection of a suitable decision model. However, the development of mechanisms that enable them to express their preferences is necessary. The type of control over the decision process is also important. The group members may follow a democratic process in order to reach a solution (e.g., voting) or they may follow a hierarchical model in which the system is supported by a mediator, capable or not of imposing decisions to the other members.

A group decision environment with a decision model will help healthcare professionals and patients clarify their position in the decision making process and assure not only that their perspectives and preferences are heard, but also that they conform with the recommendations of CPGs,

3.2 Expert systems

An expert system (ES) is a computer program capable of performing at the level of a human expert, or above it, in some knowledge domain (Nikolopoulos 1997). This type of systems uses knowledge and inference procedures to solve difficult problems. They have to mimic the adaptation capabilities of human beings in order to find solutions to new problems (Jackson 1990). In this sense, there is four fundamental aspects of the construction of ESs (Nikolopoulos 1997): the knowledge acquisition module, the knowledge base, the inference engine and the interface. The knowledge representation in an ES applies concepts of logics to create structured formalisms, inference rules as well as ontologies to define the context of the domain. The knowledge itself may be introduced in the system by human experts as rules, obtained from past experience through learning algorithms or both.

Expert systems not only apply knowledge to situations but also generate new knowledge for new situations. The advantage of these systems, namely in healthcare, is that they are able to justify the decisions they make and provide confidence measures in their decisions.

One of the problems healthcare professionals are faced with is the efficient use of all the information concerning clinical cases that they have. ESs provide means to treat large amounts of information and extract knowledge to be used in the future.

The applicability of ESs in healthcare has been proven through cases such as those of MYCIN (Melle 1978), for the diagnosis of infectious diseases; and NED (Zhou et al. 2002), which is used for the detection of lung cancer cells in the images of the specimens from needle biopsies. The usefulness of ESs in healthcare is evidenced in the work of Seto et al. (2012) that comprises the development of a rule-based ES for the monitoring of heart failure.

The development of an ES based on CPGs will enable the implementation of guideline acquisition tools based on domain ontologies that represent the different aspects of the clinical process. Such a system will also enable referencing the evidence and clinical trials that endorse a clinical recommendation, in order to provide healthcare professionals the support they need to justify their actions. Inference rules are a fundamental feature of the system. Initially, they must be based on the available evidence, researched during the guideline development process by human experts, but afterwards, learning techniques, such as CBR, may be used to reinforce the rules of the system or offer alternatives to the recommendations of guidelines. Such an ES will also enable healthcare professionals to give their feedback of guideline recommendations, according to the outcomes of their application, producing data that may be used to improve them. The issue of guideline contextualization may also be addressed by the ES through the use of information retrieval techniques to search for news and articles that fit the scope of the health conditions addressed by a guideline. Such a feature may be useful when dealing, for instance, with flu outbreaks because the healthcare professional may consider relevant the information about new virus strains that are currently active and characteristics of the population that make them particularly vulnerable to those strains, adapting his clinical practice accordingly.

Currently, web applications are growing fast. They present some advantages to their desktop counterparts that make them the ideal support for ESs. They require no installation and updating and are accessible from anywhere on the internet. The data is stored remotely and they do not require high specs from the devices in which they run. This portability makes them accessible to low spec PCs, smartphones and tablets. The coming of age of cloud computing and mobile cloud computing will have a positive impact in the way e-health services are made available to healthcare professionals, mainly due to the pervasive access to information granted by these technologies (Dinh et al. 2011). Moreover, a cloud-based health information system eases the integration of different services from different service providers through the internet to meet user demands. A web-based ES for the application of CPGs would allow healthcare professionals to access the information they need when they are in contact with their patients, filling in any knowledge gaps they might have. It would also provide decision support during the clinical process and solve the problem of guideline delivery to healthcare professionals.

Although there is a widespread research in the field of healthcare ESs, their application in real life is not so widespread. Among the reasons for this situation is the fact that, in many of them, developers do not consider the cognitive necessities of healthcare professionals when designing their interfaces (Johnson and Turley 2006). Intelligent interfaces reflect users' goals, tasks and processes in order to make human-machine interaction a collaborative experience. As such, they provide an abstraction level of the processes that occur in the internal structure of the system that resembles the cognitive process of the users. This is beneficial to the implementation of CPGs, sit enables the development of user-friendly tools consisting of graphical interfaces that support primitives for drawing the control information within the guideline, windows for acquiring the internal properties of the objects, facilities for browsing CPGs and an environment for consistency checking of clinical recommendations. To achieve these purposes, the interface needs specific data about the clinical domain that is being addressed as well as models for the representation of the knowledge of CPGs, their rules and processes.

3.3 Case-based reasoning

CBR is an AI approach that makes use of past experience to solve current problems (Aamodt and Plaza 1994). The applicability of CBR in health sciences is vast, given the similarities this research method has with the cognitive process of healthcare professionals: it is a natural process for them (Bichindarits and Marling 2006). Case histories are the main training tool for clinicians and the medical literature is filled with accounts of treatments of individual patients. Moreover, some diseases still remain a mystery to the medical community, which impedes the definition of generic models to manage them. The approach to these clinical cases requires background knowledge recorded in practice cases. These background cases complement guidelines and help to interpret them. The human body is a complex biological system that is difficult to describe and even in well-known health conditions (e.g., hypertension and heart disease) several diagnoses interact to produce a given set of symptoms.

Typically, a CBR process is composed of four sequential phases: retrieve, reuse, revise and retain (Aamodt and Plaza 1994). The first phase consists in retrieving one or more previously experienced cases that are relevant. The relevance of the cases corresponds to a similarity measure, (e.g., the difference of the sums of the different attributes that build the case). During the reuse phase, the solutions of the retrieved cases are mapped to the new case, which may involve adapting the solution in order for it to fit specific requirements of the new problem because it is unlikely that a an exact match of the new case exists in the case memory. In the third phase, revise, the best matching solution is tested in order to predict the results of its application. If the result does not meet the expectations, the action taken is revised. In the last phase, retain, the solution of the new case is stored in the case memory, contributing to its enrichment.

Among the applications of CBR systems in the health sciences domain, CASEY is one of the earliest (Koton 1988). This system diagnosed heart failure patients by comparing them to earlier patients whose diagnoses were known. CASEY also integrated an earlier model-based system and pioneered the combined use of CBR and another reasoning methodology. PROTOS (Bareiss 1989) is another early CBR system that assigned patients to pre-defined diagnoses based on past cases (Bichindarits and Marling 2006). Since the debut of these systems, CBR has been used for other tasks, such as nursing diagnosis [e.g., FLORENCE system (Bradburn and Zeleznikow 1994)], radiation therapy design [e.g., ROENTGEN system (Berger 1994)] and diagnosis of degenerative brain diseases through image segmentation of CT and MR brain images [e.g., HPISIS system (Perner 1999)], to name a few.

CBR may be used to manage non-compliance with CPGs. When executing a clinical guideline in an ES, the healthcare professional may have to face a situation that was not predicted by the guideline or in which his professional opinion is different from the recommendations provided by it. Moreover, the unavailability of relevant patient data or resources, and the existence of data that is outside the range foreseen by the guideline may also require a deviation from the protocol by healthcare professionals. When faced with these situations, the ES may allow the healthcare professional to change the guideline in order to fit the current case. With the help of CBR, the system may construct a case memory of these deviations where the description of the cases (the pair attribute/value of the clinical parameters) and their solution (the alteration made to the guideline) are stored for later retrieval to solve similar cases. This way, the system could grasp the constraints (social, economic and cultural) of the medical practice of physicians and provide useful feedback of the applicability of a certain guideline. An elevated number of cases in the memory case for a specific guideline are an indicator that a step of the guideline or the guideline itself is no longer fit for medical application.

3.4 Medical ontologies

In the context of AI, an ontology is a formal representation of knowledge as sets of concepts and the relations between them within a domain (Gruber 1993). An ontology defines a vocabulary that contains all the concepts that may be used to model the domain and how they relate to each other. This conceptualization is achieved through the definition of classes and subclasses of individuals along with the properties of the individuals in a class.

Ontologies have a key role in the Semantic Web (Berners-Lee et al. 2001),since they structure underlying data for the purpose of comprehensive and transportable knowledge and machine understanding. Besides allowing machines to read and interpret information, ontologies present other advantages to knowledge engineering such as automated validation and consistency checking.

In a complex domain such as the clinical one, ontologies provide significant advantages in the formalization of CPGs. The vagueness and ambiguity that, sometimes, is present in guidelines can be removed through the usage of controlled vocabularies, thus eliminating fuzzy relations between the concepts of the domain. It would also allow the extraction of rich patterns, that would go unnoticed otherwise, and the construction of inference mechanisms in the domain. The guideline ontology can be shared in ontology repositories for widespread use and dissemination. Currently there is a growing interest of clinical guideline researchers in ontology-driven execution of CPGs (Isern et al. 2012).

The Unified Medical Language System (UMLS) (Bodenreider 2006) reflects the efforts of the US National Library of Medicine to remove ambiguity and vagueness from the clinical setting. It is an ontology that aggregates terms used to describe the same concept from existent knowledge sources (e.g., SNOMED CT, LOINC, ICD-10, MeSH) under the same identifier. The UMLS has three main components: the Metathesaurus, the Semantic Network and the SPECIALIST Lexicon. The integration of this ontology in a CPG ontology would effectively improve the understanding of clinical recommendations and provide an easy access to semantic networks that could provide precise definitions of medical terms.

3.5 Reasoning under uncertainty

AI provides some techniques that deal with uncertainty and incomplete information in decision making scenarios. They can be classified in qualitative and symbolic methods (Clark 1990). The advantage of symbolic methods is that they bring some common sense validity that can be found in approaches such as non-monotonic logics, default logic and defeasible reasoning. There are also other symbolic methods, often called *reason-based* (Fox et al. 2001), that use informal endorsements for multiple options and formalizations of everyday strategies for reasoning about competing beliefs, *argumentation* being one of these techniques. However, the health sciences are, currently, more interested in the numeric methods such as Bayesian Networks, Dempster–Shafer Theory or Fuzzy Logic (Clark 1990).

Bayesian Networks were derived from probability theory and appeared for the first time in the middle of the 1980s (Pearl 1986). It is a statistical model defined by two components: a qualitative component and a quantitative component (Clark 1990). The qualitative component is an acyclic orientated graph whose nodes represent a random variable that may assume any value from a given set and is associated with a probability distribution. The existence of an arch between two variables means that they are statistically dependent. The quantitative component is a conditional probability distribution. The essence of this approach is the representation of hypotheses and relations in the domain under consideration. In the medical domain, the relation of causality between clinical parameters and diseases may be effectively represented through Bayesian Networks and it is possible to obtain these relations from CPGs (van Gerven et al. 2008). Moreover, the prior probabilities for the different variables and the conditional probabilities may be gathered from the empirical evidence displayed in the guideline. Thus, the combined used of Bayesian Networks and CPGs adds value to the clinical process and provides quantitative measures that enable healthcare professionals to assert the solidity of their decisions. The work of Lucas (2004) is heavily focused on the combined use of clinical guidelines and Bayesian networks for clinical decision support systems.

The Dempster–Shafer Theory of evidence was initially developed by Dempster in 1967 (Dempster 1967) and later extended by Shafer in 1976 (Shenoy and Shafer 1986). It relies in degrees of belief to represent uncertainty. This approach allows the assignment of degrees of belief to sets of hypotheses (e.g., {gastric cancer, gastric ulcer}, i.e. gastric cancer is caused by a gastric ulcer) instead of individual clinical parameters, like Bayesian networks. For this reason, it is considered that Dempster–Shafer Theory is better able to represent the process of narrowing hypotheses with the accumulation of evidence. This process is claimed to mimic diagnostic reasoning. Since Dempster–Shafer Theory identifies a set of solutions that reflect any other options that are not discretized, it can deal with ignorance and non-exhaustiveness (not pointing out all the existing solutions). However, it receives some criticism concerning the computational complexity that generates for large sets of hypotheses (Clark 1990). Despite these shortcomings, Dempster-Shaffer Theory has been used efficiently for the representation of medical knowledge and uncertainty in some critical areas (Straszecka 2004).

The Fuzzy Sets approach was initially developed with the objective of quantifying imprecise classes in natural language (Zadeh 1975). It is most useful when sets are defined by vague concepts and variables are continuous (e.g., height, warmth, age). Natural language is full of examples of fuzzy classifiers, like predicates (e.g., small, large, young), quantifiers (e.g., most, many, few), probabilities (likely, unlikely) or truth values (e.g., very true, quite true, mostly true). The quantification in this method is provided by membership functions that attribute a value in the in the interval [0,1] to the relevant elements. Fuzzy Logic was derived from fuzzy sets and is based on the notion of truth degree of a preposition. It defines operators that express the disjunction and conjunction of prepositions, independently of their meaning. Just as it is difficult to estimate the prior probabilities of a Bayesian Network, the production of membership functions is complex (Clark 1990). Many disciplines of medicine already use Fuzzy Sets in ESs, for tasks such as diagnostic and imaging analysis (Abbod et al. 2001). Fuzzy Sets are being researched for the representation of operational guideline knowledge and the definition of threshold values for clinical parameters. In fact, currently, Fuzzy Logic is being integrated with the Arden Syntax (Vetternlein et al. 2010) guideline model in order to produce a continuously graded applicability of statements.

All these approaches deal with uncertainty from a perspective of causality and interdependence, but do not address the aspects of incomplete information and the different forms it can assume. Further in this paper we will present a methodology called Quality of Information (QoI) (Neves et al. 2012) that provides ways of dealing with this information and making it useful to the decision making process.

4 Computer-interpretable guidelines

4.1 Living guidelines

As a response to the challenges presented by CPGs, the concept of computer-interpretable guideline emerged (De Clercq et al. 2004). CIGs are representations of CPGs in a digital format, suitable for being interpreted by machines. A digital format of CPGs may be a game changer in all the aspects that revolve around them, namely development, dissemination, implementation and execution.

There is a set of features that guideline researchers would like to see guidelines acquire (Rosenbrand et al. 2008). Features such as modularity, dynamism and interactivity are gathered under the concept of *living guidelines* (Seyfang et al. 2007), which basically is translated into guidelines that are easy to update and modify and have an active role in providing knowledge to healthcare professionals. The objective of researchers is to change the static and passive nature of guidelines. CIGs are, currently, the best way to achieve this purpose.

The development of a standard model of CIGs may provide a deeper understanding of the clinical process and may have significant benefits. A depiction model for CPGs can be used to identify the different requirements that must be accomplished before making a decision, to establish decision criteria and thus helping healthcare professionals in this critical moment of the clinical process (Elkin et al. 2000). Having a model also enables the definition of methods to verify the semantic and syntactic structures of guidelines, providing a way to distinguish a well formed guideline from a poorly made one (Elkin et al. 2000). If the model enables the definition of modular components, like for instance clinical procedures that are common to different guidelines, it may be possible to reuse this knowledge when building a new digital guideline (Elkin et al. 2000).

The creation and use of CIGs offer a better description and recording of patient states and may provide selective access to background knowledge to be used in specific circumstances. The use of automatic reminders according to the recommendations of guidelines may also be implemented (Fox et al. 2008).

4.2 The document-centric and model-centric approaches

Decision support systems based on CPGs may support healthcare professionals in following the best clinical practice in a consistent way. Formalization of guidelines in a guideline representation language may follow two different approaches (Sonnenberg and Hagerty 2006): document-centric and model-centric.

The document-centric approach (Kaiser and Miksch 2009) consists in using mark-up tools on the original guideline documents. The original document is either marked up or annotated to produce a more structured format with defined semantic elements. Usually, this process is carried out in stages. First, the mark-up is used to identify elements in the text of the guideline. Then, using a specialized tool, a semantic tag is assigned to the elements and the connections between them are made. The advantage of this approach lies in enabling the encoding of CIGs without the need to have a profound knowledge of a specialized computer language. However, the current tools that perform this process are not perfected yet and it is difficult to construct long and complex guidelines using this method.

On the other hand, in the model-centric approach (De Clercq et al. 2004), a depiction model is formulated by domain experts and the relationship between the new model and the original document is indirect. The acquisition of guidelines in the model-centric approach is done directly by healthcare professionals into the new model. Through this process, it is possible to develop friendlier interfaces for healthcare professionals to encode their guidelines and, at the same time, they become more knowledgeable of the different steps that compose the clinical process.

GEM Cutter (Shiffman et al. 2000), Stepper (Ruzicka and Svatek 2004) and DELT/A (Votruba et al. 2004) are some of the most relevant mark-up-based tools that generate semi-formal models of marked guideline texts. GEM Cutter was one of the first tools to apply a document-centric approach and transform guideline information to an ad hoc format, called GEM. Stepper is a tool that segments the initial text in multiple user definable steps coded in XML. DELT/A stands for Document Exploration and Linking Tool/Add-ons and, as its name indicates, it supports the translation of HTML documents into any XML language, among which is the Asbru guideline representation model. There are methodologies [e.g., LASSIE (Kaiser and Miksch 2009)] for document-centric approaches that use information extraction techniques that rely on databases of medical terminologies to acquire guidelines in a semi-automatic way, thus eliminating the requirement of having an healthcare professional manually tagging the terms in the original document.

Some applications for model-centric acquisition of CPGs will be presented when the different representation models for CPGs are addressed, further in this paper.

4.3 Aspects of CIG-based systems

In the conception and development of CIG-based decision support systems, researchers identified four aspects that must be taken into consideration in the development process (De Clercq et al. 2004): guideline representation and modelling, guideline acquisition, guideline validation and testing, and guideline execution.

The model is the fundamental feature of a CIG-based decision support system (Peleg et al. 2003). It has to provide enough expressivity in order to accommodate every step of a guideline. Normally, the models created exclusively for guideline representation have a set of construction units (e.g., tasks or steps) that are used to build a guideline (De Clercq et al. 2004). These building blocks are given the designation of representation primitives (e.g., decisions, actions) and are used according to a Task Network Model (TNM). Some works consider the adaption of business process models, such as Petri Nets (Quaglini et al. 2000), to the modelling of CPGs. However, these approaches do not have enough expressivity due to them being developed to support business organizations and processes rather than medical organizations and processes. The possibility of using them in clinical settings is being actively studied in order to define higher abstraction layers, capable of expressing the different steps of the clinical process, on top of the basic model.

Whichever model is chosen, the degree of complexity the representation is able to accommodate is an important factor. Different models may differ in terms of the abstraction levels they allow, for instance, in the nesting of guidelines inside other guidelines. CPGs possess two different types of knowledge (Rosenbrand et al. 2008), the declarative (scientific knowledge about the domain) and procedural (the inference methods and the decision model), which have to be formalized through a language in the representation model. The language should provide an objective vocabulary, syntax and semantics, so that an inference engine can be developed. In a complete representation, there should also be triggering criteria, which include initial screening to assess if a patient should enter a protocol or not and connect the different elements of the guideline according to the output of decisions. Another indispensable feature is temporal patterns because guideline recommendations depend mostly of the state of the patient in a given moment. Knowing this, it is essential for a guideline model to provide mechanisms to define durations, repetitions and cycles of tasks.

An acquisition tool must be developed in order to help healthcare professionals structure the knowledge according to the guideline model that was defined (De Clercq et al. 2004). The tool must take into account the approach followed for guideline acquisition, if it is either document-centric or model-centric.

The precision, the syntactic correctness and the semantic coherence are extremely important in the acceptance of CIGs by healthcare professionals and in their integration in daily practice (De Clercq et al. 2004). As such, the inclusion of mechanisms for guideline validation and testing in the guideline execution engine is necessary. During the execution, the guideline execution engine should have access to a database containing the values for the clinical parameters that build the patient's state in order to apply CPGs in real time.

4.4 Current approaches to guideline modelling and execution

Currently there are few CIG systems available and they lack application in real clinical settings. This review addresses them by their depiction models and mentions the execution engines available for each one as well as the underlying platforms. The selection of the approaches was based on opinions collected from the literature that deemed them the most relevant. Table 1 shows a summary of the available software tools and models as along with their main features.

4.4.1 Arden Syntax

Arden Syntax (Hripcsak 1994) was developed in 1989 and is now a standard of Health Level 7 (HL7). The current version of Arden Syntax is Arden Syntax 2.8. The primary aim of this approach is the sharing of simple and independent guidelines as modules. Each clinical guideline is modelled as a medical logic module (MLM), which comprises relevant knowledge for only one judgment. Initially, each MLM was an ACII file divided in three partitions: *maintenance*, *library* and *knowledge*.

The *maintenance* and *library* partitions possess administrative information about the guideline, namely authoring and version number. The constructs of the *maintenance* partition are *title*, *(file)name*, *author*, *version*, *institution*, *date of last modification* and *validation status*. The *validation status* contains information about the approval of the guideline in a local institution and it may have three possible values: *testing*, *research*, *production* and *expired*. The transition from *testing* to *production* means a shift of responsibility from the institution that developed the MLM to the local institution where the guideline will be applied. The *library* compartment contains constructs used for a detailed description of the guideline and among them the attribute *purpose* enables the expression of the clinical objective of the MLM.

The main constructs of the *knowledge* compartment are *data*, *evoke*, *logic* and *action*. The *data* construct is used to obtain the values of the concepts referred in the MLM from the information system of the healthcare institution. The *evoke* construct contains the events that trigger the execution of the MLM and these events are related with the clinical parameters in *data*. The decision criteria are expressed in the *logic* construct through *if-then-else* rules and sets of logical, mathematical and temporal operators. When a rule is assessed to the value *true*, a given procedure of the construct *action* is proposed. These procedures may include messages/alerts or the execution of other MLMs. This approach reveals great modularity and gives transparency to the decision making process, but given its simplicity, the ability to capture the full content of a clinical guideline is compromised. Arden Syntax is mainly used in alert and monitoring systems, like the ones provided by the Regenstrief Institute (Anand et al. 2004).

Initially it was defined in Backus-Naur Form (BNF), a notation technique used to describe the syntax of computation languages. Currently the development of Arden Syntax by HL7 is based on XML (Kim et al. 2008).

There is a myriad of tools to acquire and execute guidelines in Arden Syntax. We will highlight the Arden Syntax IDE (Samwald et al. 2012), which is a simple development environment that provides syntax highlighting and testing functionalities for MLMs. The Arden Syntax IDE contains a compiler that generates java classes from MLM code. These classes are then executed by an *Arden Syntax Rule Engine* that works together with another component, the *MLM manager*, which gives the rule engine the access to the available MLMs in the system. Arden Syntax is a highly portable format, conceived to be integrated in Clinical Management Systems (CMSs).

4.4.2 Guideline interchange format (GLIF)

The guideline interchange format (GLIF) (Ohno-Machado et al. 1998) represents an effort of Intermed Collaboratory for the development of a sharable clinical guideline representation model. The first published version of GLIF was released in 1998 and its current version is GLIF3 (Boxwala et al. 2004). This approach was developed in order to reflect a flowchart of steps and consists of a set of classes that describe the fundamental characteristics of a guideline and constructs that contain the clinical parameters. Through this flowchart representation, GLIF3 provides a better understanding of the clinical process to healthcare professionals.

A guideline in GLIF3 is an object that contains different steps, namely: *decision steps*, *patient state steps*, *branch steps*, *synchronization steps* or *action steps*. This approach follows the Task Network Model (TNM), so that every moment of the clinical process is labelled as a step.

Decision steps model decision points in a guideline and direct the careflow from one to alternative steps. There are two subclasses in *decision*: *case step* and *choice step*. A *case step* contains a set of logical expressions that initially corresponded to an excerpt of Arden Syntax. Currently, GLIF3 uses an OCL (Object Constraint Language) expression language called GELLO (Sordo et al. 2004) that has more expressive power than the previous. As for *choice steps*, they contain only a set of options for the next step in the clinical process and the selection is made by an external agent (e.g., the user).

Patient state steps function as labels that have constructs used to describe the patient's health condition. These steps are used as data entry points in the system. When the state of the patient is updated, the guideline that possesses the corresponding patient state is executed.

The tasks of the clinical process are modelled in the *action steps* through three distinct constructs: *medical actions*, *activity oriented actions* (e.g., messaging, retrieving of patient data) and *control actions* (invocation of structures such as sub-guidelines).

At Columbia University, GLIF is being integrated with the Clinical Event Monitor and the Computerized Physician Order Entry systems to provide clinical decision support (Peleg and Wang 2006) for post-CABG (Coronary Artery Bypass Grafting) (Zheng et al. 2010). Encoded GLIF guidelines are also being used in Israeli clinics for the management of feet injuries of diabetics. The GLIF3 Guideline Execution Engine (GLEE) (Wang et al. 2004) is a tool for executing guidelines in this format. It defines three layers of abstraction: *data*, *business logic* and *user interface*. The*data* level contains the Electronic Medical Record (EMR) and a guideline repository. The execution engine is in the *business logic layer* and includes a server that interacts with the data layer and clients that interact with the users. Applications exchange data with the other two layers through the *user interface* layer. GLEE may be linked with a clinical event monitor, thus enabling event-driven execution of CPGs, responding to alterations in the state of the patient. This software tool also defines a set of methods to connect it to CMSs and uses representations like the Resource Description Framework (RDF) and HL7 as a general patient data model (Schadow et al. 2006) to support CPGs and encode medical data in order to share information across different institutions.

4.4.3 PROforma

In 1998, the Advanced Computation Laboratory of Cancer Research of the United Kingdom initiated the development of the PROforma (Fox et al. 1998; Sutton and Fox 2003) depiction model. The objective of this model is the development of guidelines as flowcharts where the nodes are instances of pre-defined classes of tasks. The main classes are *plans*, *actions*, *decisions* and *enquiries*. Each class has a set of attributes that reflect its information needs. The syntax of PROforma was also initially defined in BNF in an ASCII file.

Every task of a guideline derives from a common task called *root task*. A *root task* contains several guidelines encoded as sets of tasks called *plans*. On the other hand, a *plan* contains any number of instances of atomic tasks, such as *actions*, *decisions* and *enquiries*. A *plan* also has constructs that enable the definition of clinical objectives (that reflect the objective of a guideline), abort or termination conditions and scheduling constraints on the atomic tasks. It is also possible to define temporal constraints on *plans*, such as cycles, durations and number of repetitions. One of the core features of PROforma is the possibility of nesting plans inside other plans.

An *action* in PROforma is a task whose execution has to be performed by an external agent. Typically, these tasks consist in sending messages and calling external programs or clinical procedures.

The *enquiry* task defines data entry points in the guideline as questions to the patients or internal procedures to retrieve the relevant information from the patient's EMR. This class contains *data definition* constructs that specify how a value for a clinical parameter must be stored (e.g., data type, unit).

Perhaps the most important class in PROforma is *decision*. Among all the CIG formalisms, PROforma was the first to offer a support to deal with uncertainty during the decision process. A *decision* contains constructs to express candidate solutions to the decision problem as well as logical expressions that endorse or refute each candidate. Each expression, in favour or against a candidate, is associated with positive signs (represented by a plus sign $+$) and negative signs (represented by a minus sign $-$). The weight of an argument in the overall score of its candidate depends of the number of positive and negative signs it has. This is a symbolic method of endorsement that uses a mathematical function to convert the signs in numeric values for calculating the scores of each option. Then the options are presented by descending order of scores. According to the results of a *decision* task, the next task in the careflow is the one that has a construct called *trigger condition* matching the output of the *decision*.PROforma has been used in a few prototypes for clinical management, namely CAPSULE for general practice and Bloedlink for advice on laboratory tests and management of chronic diseases (dyspepsia, asthma and depression) (Fox and Thomson 1998).

Among the software tools for PROforma, Arezzo (Fox et al. 2006) is arguably the most disseminated. It has an architecture composed of three elements: a *composer*, a *tester* and a *performer*. The *composer* is responsible for providing an acquisition suite of guidelines in PROforma. The *tester* verifies the syntactic integrity of the PROforma guidelines before deployment by the *performer*, which is an inference engine. The *performer* can be linked to existing EMRs and CMSs to acquire data related to patients and also defines different states of guideline execution (e.g., *waiting for data*, *suspended*, *finished*).

Table 1 Software tools for guideline development (adapted from Isern and Moreno 2008)

Tool	CG repository	CG editor	CG representation language	CG basic elements	Run-time engine	Access to EMR	Access to CMS	Standards used
Arden Syntax IDE	Yes	Yes	Arden Syntax	MLMs	Rule-based	No	No	XML
GLEE	Yes	Yes	GLIF3	Decision, action, patient state, branch, synchronization steps	Event-based and rule-based	Yes	Yes	RDF,HL7
Arezzo	Yes	Yes	PROforma	Plan, action, decision, enquiry	Rule-based	Yes	Yes	No
DeGeL	Yes	Yes	Asbru	Preferences, intentions, conditions, effects, plan body	Rule-based	Yes	Yes	XML, ICD-9, SNOMED-CT, CPT, LOINC
GLARE	Yes	Yes	Graph-like	Query actions, work actions, decision actions, conclusions	Rule-based	Yes	Yes	XML, ICD-9
SAGE	Yes	Yes	SAGE model	Context, action, decision, routing nodes	Event-based	Yes	Yes	HL7, UMLS

4.4.4 Asbru

Asbru (Shahar et al. 1998) is the result of collaboration between Stanford and Vienna Technology Universities. This formalism presents a notion of *plan* similar to PROforma in the sense that it represents a collection of items. The knowledge required to perform a *plan* is defined by its *knowledge roles*, which include *preferences*, *intentions*, *conditions*, *effects* and *plan body*.

The content of a *plan body* is composed of other *plans* until they are no longer decomposable, reflecting a parent-child structure. The plans that cannot be decomposed are called *actions*. The functionalities of *plans* and *actions* are defined by the remaining *knowledge roles*. The *plan body* is the layout of a given *plan*.

The restrictions on the execution of a *plan*, in order to achieve a given objective are defined by *preferences*. The categories in preferences that define these restrictions are *select-method*, *resources* and *strategy*.

The objectives of *plan* are represented in the *intentions knowledge role*. The definition of *intentions* helps the selection of an adequate *plan* and is crucial in decision support. Intentions are defined as temporal constraints on the actions of healthcare professionals. There are four types of intentions: *intermediate state*, *intermediate action*, *overall state pattern* and *overall patient pattern*. *Intermediate state* refers to patient states that must be maintained, reached or avoided (e.g., the control of levels of substances in the blood) during the execution of the *plan*. On the other hand, *intermediate actions* define the actions the healthcare professionals must perform during the *plan*. The *overall state pattern* is the state of the patient that must be verified at the end of the execution of the *plan* and the *overall action pattern* is the pattern of actions of the healthcare professional that should result from the *plan*.

There are different types of *conditions*, namely *filter-preconditions* and *setup conditions*, *suspend conditions*, and *abort conditions*, that are used to express the respective following situations: conditions that must hold for a *plan* to be considered applicable, conditions that determine the suspension of a *plan* and conditions that determine the abortion of a *plan*.

Effects describe the expected behaviour of the execution of a plan. It is composed of the following two constructs: *argument-dependency* and *plan-effect*. The first is used to describe the functional relationship between the *plan* arguments and the measurable parameters, describing how they influence each other. The second describes the relationship between the overall plan and its expected effect.

Asbru is heavily focused on temporal aspects of CPGs and that is evident in its temporal annotations, which specify four points in time for the execution of plans and verification of conditions, with the particularity of allowing the expression of uncertainty in starting time, ending time and duration of a time interval. The temporal annotations of Asbru are *earliest starting shift* (ESS), *latest starting shift* (LSS), *earliest finishing shift* (EFS) and *latest finishing shift* (LFS). It is also possible to specify two types of durations: minimum duration (MinDu) and maximum duration (MaxDu).

This model has been used in the Asgaard project in the development of prototype applications for the management of diabetes, jaundice and breast cancer (Zheng et al. 2010).

Acquisition and execution of Asbru guidelines is possible through DeGeL (Shahar et al. 2004), a tool in development at Ben Gurion University, in Israel, and is a web-based architecture that facilitates the conversion of textual guidelines to Asbru guidelines. This distributed architecture has some key components responsible for the creation of new guidelines, guideline retrieval from an XML repository as well as testing and execution of guidelines. The execution module is called *Spock* and it incorporates an inference engine that can retrieve data from the patient's EMR. It is a modular client-server application that consists of a set of classes to store guidelines, a parser to interpret their content and a synchronizer that establishes the communication with external systems. DeGeL also has a vocabulary server for supporting guideline specification and establishing mappings between the standardized terms and each clinical database vocabulary. The set of standard terminologies that is used includes International Classification of Diseases (ICD-9), Standard Nomenclature of Medicine (SNOMED-CT), Current Procedural Terminology (CPT) and Logical Observation Identifiers, Names, and Codes (LOINC).

4.4.5 GuideLine Acquisition, Representation and Execution (GLARE)

The GuideLine Acquisition, Representation and Execution (GLARE) (Bottrighi et al. 2006) project includes a guideline depiction model and a system to acquire and execute CPGs. It was developed by the Computer Science Department of the University of Piemonte Orientale, Alessandria, Italy.

The depiction model does not use a standard representation. Instead, it defines a proprietary graph-based structure for displaying CPGs, where a clinical action is represented by a node. It is possible to define atomic actions that represent simple tasks like *queries* to obtain external information, w*ork actions* that represent medical procedures, *decision actions* with sets of conditions to select alternatives and *conclusions* that describe the output of a *decision*. *Decision actions* are specific types of actions that contain the criteria used to select from alternative paths from a guideline. These criteria are represented as sets of triplets in the form ⟨diagnosis, parameter, score⟩ and, in turn, a parameter is another triplet ⟨data, attribute, value⟩. It is also possible, in GLARE, to define composite actions, which are collections of atomic actions or other composite actions. GLARE was designed to cope with different types of temporal constraints and implements specialized temporal reasoning algorithms.

The GLARE execution engine (Bottrighi et al. 2006) distinguishes between the acquisition phase and the execution phase of guidelines. Similarly to GLEE, GLARE defines three architecture levels, namely S*ystem*, *XML* and *DBMS*. The *System* level encompasses the acquisition and execution modules. The *XML* level is responsible for the data exchanges between the *System* level and the *DBMS* level. The *DBMS* is the bottom level, responsible for establishing a physical connection between the top levels and the databases where the information for creating and executing guidelines is stored. This information includes open instances of guidelines, a repository of guidelines and medical records of patients. GLARE uses ICD-9 as a terminology standard.

4.4.6 Standards-based sharable active guideline environment (SAGE)

The standards-based sharable active guideline environment (SAGE) (Ram et al. 2004; Tu et al. 2007) project is a collaboration of six research groups (IDX Systems, University of Nebraska Medical Center, Intermountain Health Care, Apelon, Inc., Stanford Medical Informatics and the Mayo Clinic). SAGE includes a guideline depiction model and a guideline authoring and execution environment and is perceived as an evolution of GLIF3 and EON. Its objective is to establish an infrastructure to enable sharing guidelines in heterogeneous clinical information systems. SAGE is involved with standards organizations to bridge the gap between guideline logic and real life implementations.

In this depiction model a guideline is a *recommendation set*, which is composed as a graph of nodes. These nodes can be *context*, *action*, *decision* and*routing nodes*. *Context* *nodes* describe the environment in which the guidelines are applied (e.g., a physician in an emergency room). The *action nodes* represent activities of the information system that support the execution of a guideline. The control of the careflow is performed by the *decision* and *routing nodes*. The patient state is retrieved directly from the electronic health record of the healthcare entity.

So far, application of SAGE in practice is very limited. However, the Mayo Clinic has plans to apply it in the implementation of guidelines for immunization, diabetes and pneumonia in controlled environments (Zheng et al. 2010).

The SAGE system consists of an execution engine, an event listener and a set of services (terminology, patient record and general applications) (Ram et al. 2004; Tu et al. 2007). The execution engine is called *SAGEDesktop* and it interprets the content of the *context*, *action decision* and*routing nodes*. The event-listener communicates with the CMS and the EMR with the objective of detecting sudden alterations in a patient's state and notifies the execution engine if that is the case. There is also a terminology server that was added to customize terms used in local applications. The communication between the CMS and the execution engine is facilitated by and Application Programming Interface (API) developed specifically for this purpose, which hints to the main focus of SAGE, interoperability. Semantic and syntactic interoperability of clinical data requires the use of standard data types, terminology, information models and conventions for expressing clinical statements. SAGE puts this to practice through the use of HL7 version three and the UMLS.

5 Active guidelines in a clinical setting

After the analysis of some of the existing projects in the field of CIGs, there are some issues that leave room for improvement and innovation. This work focuses mainly on guideline modeling and decision support during guideline execution.

Concerning guideline modelling, there are some issues that may be pointed out, namely the fact that the available models lack real life application outside the academic environment and are still in the development phase. As such, there is not a reference standard for CIG representation that can be used when developing a system for acquisition and execution of CPGs. None of the models was largely adopted by the health informatics community. The degree of complexity the different models can accommodate is also a matter of discussion: the model cannot be too simple because it may not be able to represent all the information contained in a guideline. A paradigmatic example of this case is Arden Syntax, arguably the model with the highest number of implementations, with its MLMs capable of only representing a decision point in a guideline. At the other end is PROforma, which defines a number of proprietary specifications for data that may be difficult to use and apply to real practice. The challenge is to develop a model capable of representing complex guidelines, yet simple enough to do it with a minimal set of components. Most models and tools do not use terminology and data model standards, which makes the transition to clinical applications in a clinical setting difficult and impairs interoperability with other software tools already used in such environments. Moreover, some of the models require some proficiency in languages to formalize logical rules, numerical expressions and temporal constraints (e.g., BNF, GELLO), which non-expert clinicians do not have, thus precluding the correct acquisition of CPGs. Furthermore, the software tools provided for editing, visualizing and executing guidelines are often too complex and not user-friendly. The ontology and related tools should be developed in order to allow clinicians with no advanced programming knowledge to revise and customize the guideline representations. Most of the existing models are specialized in certain disease domains which limits their capability to represent other knowledge domains and their applicability to other areas of medicine.

Decision support has also been neglected in the current CIG approaches. The current systems do not complement the decision schemes proposed by their models with techniques to infer the confidence in the outputs of the decision process. Furthermore, the problem of incomplete and uncertain information mentioned in previous sections of this paper remains unaddressed.

Next one will mention the efforts that are underway towards the development and implementation of active CPGs, by extracting elements from the current CIG representations and applying the above mentioned techniques of AI. Ultimately, the aim is to create truly interactive and *living* guidelines by building upon the work done so far and through the introduction of new technologies, models and methods.

5.1 Ontology for clinical practice guidelines

The approach to guideline modelling that one intends to develop presents an abstract view of decision making processes and task management during a clinical procedure (Oliveira et al. 2012). The model is depicted in Fig. 1. The main objective is the development of an ontology capable of accommodating any clinical guideline. To do so, the model should be generic in order to adapt to the context and necessities of different guidelines and, at the same time, allow the definition of constraints characteristic of clinical workflows. There are certain aspects to take into account when developing the model, namely scheduling constraints of the recommendations, time constraints, clinical parameter constraints, terminology and the modularity of the model.

Every task described in a clinical guideline is modelled as a task displayed in an oriented graph. The task is the basic unit of the model. As so, a guideline is viewed as a *plan*, which is a collection of tasks represented by the following constructs: a*ction,* q*uestion*, *decision*, and *aggregation module*. A *plan* has any number of instances of these constructs and their ordering and sequence inside a *plan* will be expressed in the form of a linked list that connects the different instances.

Actions represent tasks that must be performed during the execution of a guideline. They can either be clinical procedures, exams, medication prescription, simple recommendations or internal data operations (e.g., calculation of the body mass index from the available clinical parameters).

To feed inputs to the system one uses the *question* task. A *question* is a task to obtain data about the patient's state, in the context of a guideline. It is a data entry point that acts as the substratum for the execution of the other tasks, it is the mechanism through which one feeds information to the CIG system. It also contains a series of constructs to describe the clinical parameters and the data types for their values.

When a decision point is reached in the guideline workflow, the decision task is used. This task contains rules that associate options to the parameters of the patient's state. It has constructs to express the conjunction and disjunction of conditions.

The *aggregation module* aims at controlling special cases in guidelines and groups tasks that are part of a cycle or iteration, creating the conditions for the user to define their *periodicity*, *duration* and *objective*. It also enables the representation of tasks that belong to alternative pathways of the clinical workflow, like the ones that follow a *decision* task, in which the system chooses the next step of the clinical process according to the conclusion reached at the *decision*. One more requirement is the representation of simultaneous tasks that should be executed in parallel.

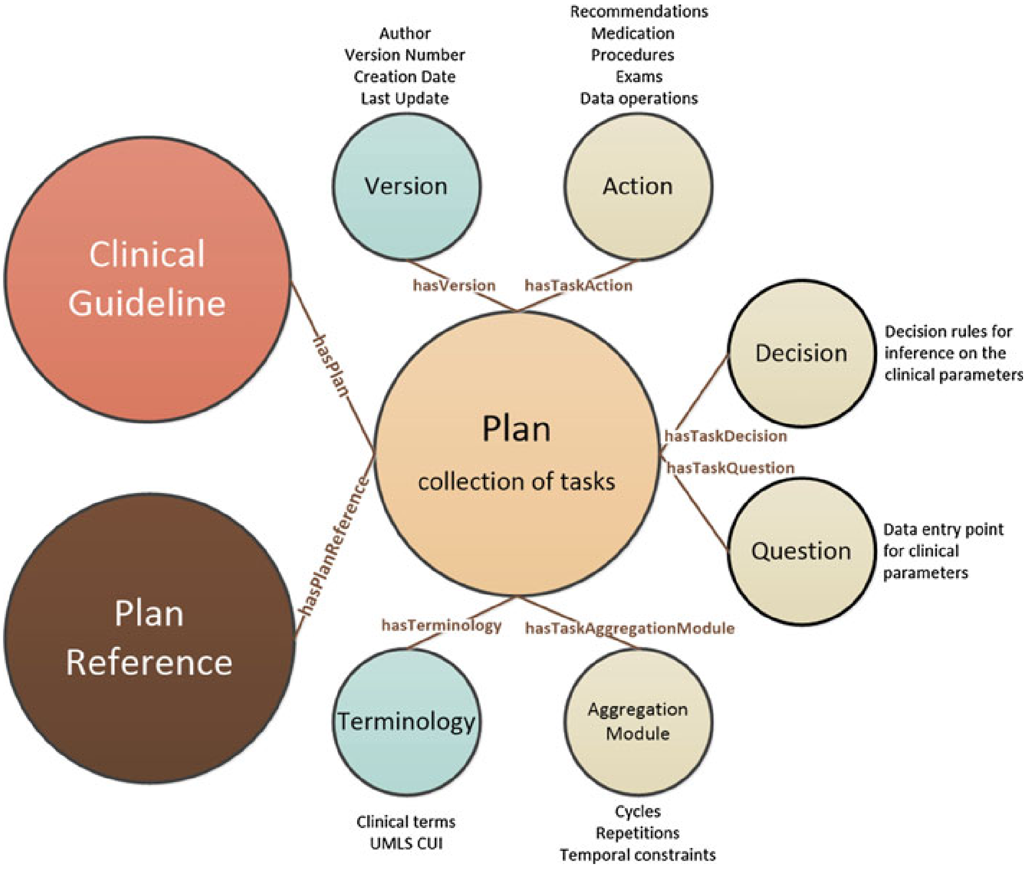


Fig. 1 Representation of the ontology proposed for guideline modelling

Another relevant aspect of the model is the *terminology* construct of a *plan*. *Terminology* comprehends the terms used in all the tasks of the plan along with their Concept Unique Identifier (CUI) (Bodenreider 2006), which is a code used in the UMLS Metathesaurus to identity a concept and associate the different terms that can be used as synonymous. This controlled vocabulary is an answer to the ambiguity and vagueness of CPGs.

The *version* construct contains administrative information about the guideline and its authoring, as displayed in Fig. 1. Also in Fig. 1, it is visible a construct called *plan reference* whose function is to make a reference to other *plan* that must be executed in the context of the current one.

To capture the knowledge of the domain and thus create the guideline ontology one will resort to the Ontology Web Language (OWL) (Antoniou and Harmelen 2009) because it is the emerging modelling paradigm of the Semantic Web and a standard of the World Wide Web Consortium (W3C). More specifically, the OWL undertaking to be used is OWL-DL, which uses description logics to formalize its classes, individuals and properties. There are a number of reasoners developed to verify the semantic correctness of OWL ontologies (e.g., HermiT, FaCT++), which is an advantage of using this language for modelling guidelines. Moreover, the underlying support for OWL is provided by RDF and XML, which are well known standards.

The set of tools that support the ontology are crucial and they greatly determine the adoption (or not) of the model by the medical community. Knowing this, the ontology will be integrated in a web-based ES, whose advantages were already mentioned in this paper. The importance of the interface in such a system is paramount, mainly because the interface is often the factor of exclusion of a system by clinicians. The system must possess a patient data model and it is essential that the data model is compliant with HL7, namely HL7 version 3 (Dolin and Alschuler 2011).

5.2 Clinical decision model

Before applying a clinical decision model that contemplates incomplete information, it is necessary to represent this information in an appropriate way. The Extension to Logic Programming (ELP) (Neves et al. 2012; Novais et al. 2010) is one of the few techniques that enable this representation, using Mathematical Logic. ELP uses two types of negation: default (weak) negation and classic (strong) negation. The use of these two types of negation is the core feature that enables the association of ELP programs to sets of abducibles, represented as exceptions to the extensions of the predicates that represent the clinical parameters. This representation technique augments the usual truth values that are assigned to information (*true* and *false*), by adding the truth value *unknown*, and allows one to represent explicitly negative information. For instance, in cases of inexactitude where there are different possibilities for the value of a clinical parameter, these possibilities are represented as abducibles or exceptions. In cases of uncertainty, if the value of the clinical parameter is unknown, this is represented as a null value.

Decision making in these situations requires the use of an information quantification method. The Quality of Information (QoI) (Neves et al. 2012; Novais et al. 2010) is a methodology associated with ELP. It is defined in terms of truth values taken in the interval [0, 1] that are attributed to the clinical parameters of the patient according to their number of abducibles and null values. Knowing this, it is possible to calculate the QoI of each condition in a *decision* and calculate scores for each option with the relative weights of its conditions.

By assimilating the concepts of CBR and contextual information with the ELP and the QoI in the context of a runnable clinical guideline, it is possible to devise a decision model that focuses on preeminent matters of guideline execution, non-compliance and inadequacy (Oliveira et al. 2012). Such a decision model is depicted in Fig. 2. Starting with the retrieval of relevant information about the clinical parameters of the patient, this data is presented to the healthcare professional along with a feed of contextual information. This newsfeed is composed of recent news and articles retrieved by an agent from relevant online sources (e.g., the website of the Center for Disease Control and Prevention). Then, based on this information, the healthcare professional assesses the adequacy of the guideline to the case in hand and defines if it is a compliance situation with the guideline he is following or, on the contrary it is a non-compliance situation. In the compliance situation, the decision process moves to the core stages of the decision model. These core stages start with the *Formulation of Clinical Hypotheses*, where the system carries out a survey on the available options in a *decision* task of the clinical guideline. The following stage is *Voting*, where, for each option and consequently for the rules that dictate their choice, the system performs an *Evaluation of Conditions*, to see if they hold true. Next, in the *Evaluation of the QoI*, the system assesses the state of the information responsible for validating each rule and assigns a score to each option. In the following stage, the *Selection of the Clinical Option*, the output of the *decision* is generated. The selected option will be used as a trigger condition for the following tasks in the clinical process. In the *Clinical Task Assignment*, the next task of the clinical process is selected according to its trigger condition. On the other hand, before a non-compliance situation the system may perform one of two things: retrieve a similar case from the case

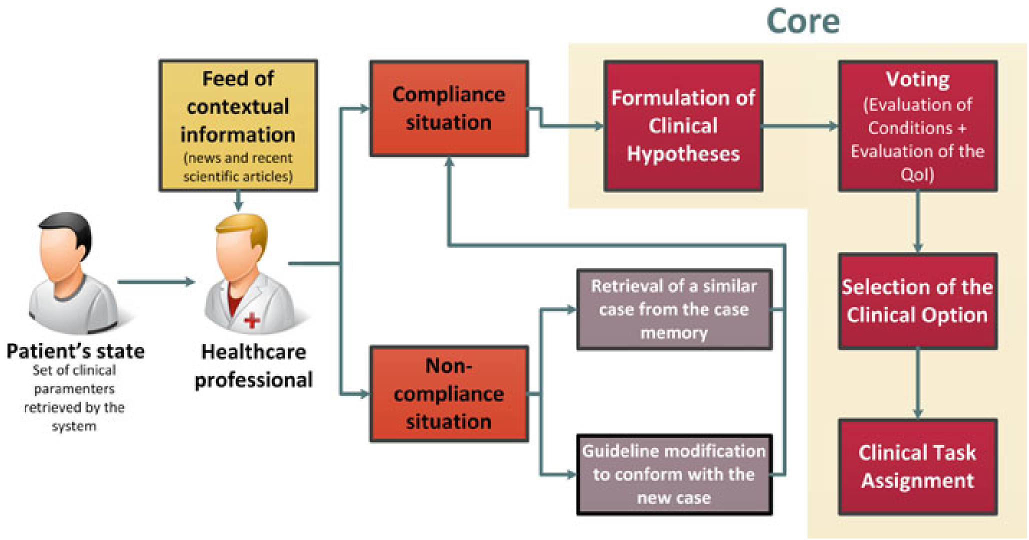

Core
Feed of contextual information
(news and recent scientific articles)
Patient's state
Set of clinical paramenters retrieved by the system
Healthcare professional
Compliance situation
Non-compliance situation
Formulation of Clinical Hypotheses
Voting
(Evaluation of Conditions + Evaluation of the QoI)
Selection of the Clinical Option
Clinical Task Assignment
Retrieval of a similar case from the case memory
Guideline modification to conform with the new case

Fig. 2 Clinical decision model for the execution of clinical practice guidelines in an expert system

memory or suggest that the healthcare professional alters the current guideline in order to fit the current case. The case memory contains the previous alterations made to the guideline, as well as the clinical parameters of the patient that made him alter it and the output of the process generated by the alteration. If a similar case does not exist, the healthcare professional alters the guideline accordingly and this alteration will enter the memory case as a new case. Once selected the case or made the alteration, the system shifts from a non-compliance situation to a compliance one and enters the core stages of the model.

This decision model leaves the door open to further research on the complementarity that other techniques that manage uncertainty in different ways, namely Bayesian Networks, Dempster–Shafer Theory and Fuzzy Logic, may offer to the QoI.

The implementation of such a decision model is necessary in order to capture the context of the execution of guidelines and provide measures of confidence in the outputs.

The development of such a decision model is but a step in the construction of a wider decision platform, represented in Fig. 2, where healthcare professionals, members of the same medical team, possibly dispersed across different locations, can discuss the case of a patient in the context of an intelligent environment. Through the use of AI techniques, it is possible to perceive information about the state of stakeholders, namely their attitudes and emotions and thus determine the type of interaction they are developing. If one throws into the equation relevant knowledge, from exterior sources, concerning the health condition (that is the object of the discussion) and guideline recommendations, a group decision environment is established for healthcare professionals to discuss if a guideline is suitable for the situation at hand and mediate/negotiate solutions. Having all this information enables the medical team, in cases of non-compliance of guidelines, to build new strategies and adapt their content to maximize the probability of a successful treatment (Fig. 3).

6 Conclusions and future work

It is widely accepted that the adoption of CIGs would greatly improve the efficiency of healthcare, both in clinical and in economic aspects. This is an on-going research line with

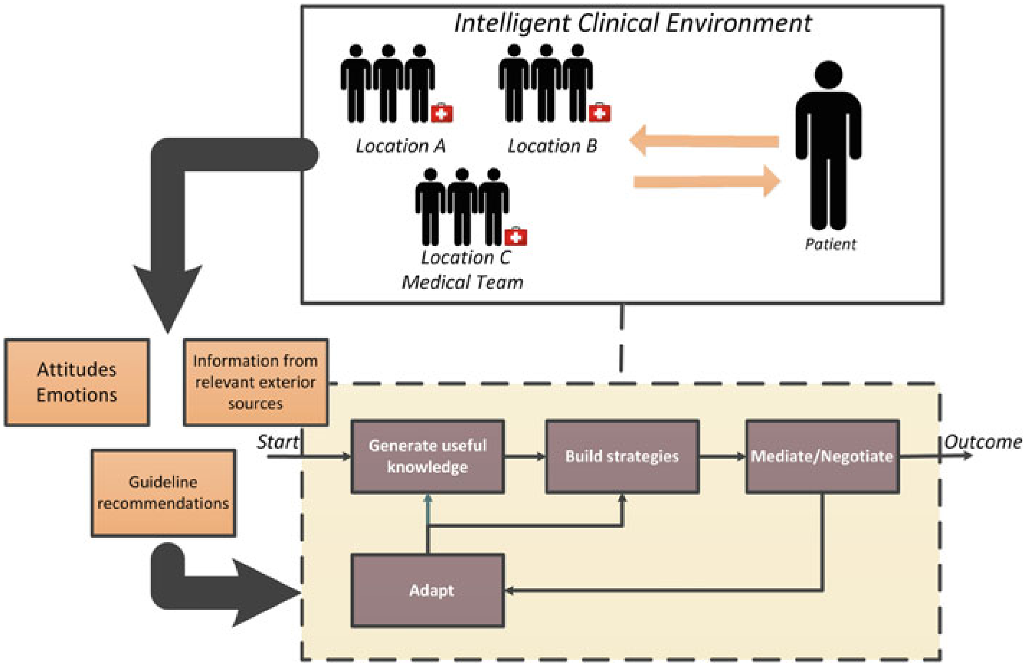

Intelligent Clinical Environment
Location A
Location B
Location C
Medical Team
Patient
Attitudes
Emotions
Information from relevant exterior sources
Guideline recommendations
Start
Generate useful knowledge
Build strategies
Mediate/Negotiate
Outcome
Adapt

Fig. 3 Characterization of an intelligent clinical environment where a group decision framework is established using clinical practice guidelines

numerous people working on the implementation of useful models and the development of execution engines. However, after perceiving the main necessities of paper-based CPGs and analysing the current CIG approaches, one may conclude that they do not solve completely the shortcomings of guidelines, as it is evident by the fact that the available models and systems are not widely implemented.

The line of research proposed in this paper focuses on the development of an ontology for the representation of CPGs that effectively encompasses different clinical domains and, at the same time, shows portability, for implementation in heterogeneous systems. The requirements to achieve this purpose include the conception of structures to accommodate different types of clinical tasks, temporal and scheduling constraints, logical rules, triggering criteria and shows conformance with data and terminology standards. The current CIG models are not complete in the way that they do not have a transversal approach to all of these issues.

It is also viable to conclude that there is a need for a decision model that addresses the aspects of the contextualization of guideline execution and the handling of incomplete and uncertain information. However, healthcare professionals and their opinions should not be excluded of the process because one of the current criticisms to guidelines is that they are too rigid and do not give space for innovation and change.

Acknowledgments This work is funded by national funds through the FCT—Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011”. The work of Tiago Oliveria is supported by a doctoral grant by FCT (SFRH/BD/85291/2012).

References

Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun 7(1):39–59

Abbod M, von Keyserlingk D, Linkens D, Mahfoul M (2001) Survey of utilization of fuzzy technology in medicine and healthcare. Fuzzy Set Syst 120:331–349

Anand V, Biondich PG, Liu G, Rosenman M, Downs SM (2004) Child health improvement through computer automation: the CHICA system. Medinfo 11(1):187–191

Antoniou G, Harmelen F (2009) Web ontology language: OWL. In: Staab S, Studer R (eds) Handbook on ontologies. Springer, Heidelberg/Berlin, pp 91–110

Anogianakis G, Maglavera S, Pomportsis A, Bountzioukas S, Beltrame F, Orsi G (1998) Medical emergency aid through telematics: design, implementation guidelines and analysis of user requirements for the MERMAID project. Int J Med Inform 52:93–103

Bareiss R (1989) Exemplar-based knowledge acquisition: a unified approach to concept representation, classification, and learning. Academic Press Professional Inc, San Diego

Barnett G, Hoffer E, Packer M (1992) DXplain-demonstration and discussion of a diagnostic decision support system. Proc AnnU Symp Comput Appl Med Care: 822

Berger J (1994) Roentgen: radiation therapy and case-based reasoning. In: O'Leary D, Selfridge P (eds) Proceedings of the 10th conference on artificial intelligence for applications. IEEE Computer Society Press, Los Alamitos, pp 171–177

Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. Sci Am 284(5):34–43

Bichindarits I, Marling C (2006) Case-based reasoning in health sciences: what's next? Artif Intell Med 36:127–135

Bodenreider O (2006) The unified medical language system (UMLS): integrating biomedical terminology. Nucl Acids Res 32:267–270

Bottrighi A, Terenziani P, Montani S, Torchio M, Molino G (2006) Clinical guidelines contextualization in GLARE. AMIA Annu Symp Proc: 860

Boxwala A, Peleg M, Tu S, Ogunyemi O, Zeng Q, Wang D, Patel V, Greenes R, Shortliffe E (2004) GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. J Biomed Inform 37:147–161

Bradburn C, Zeleznikow J (1994) The application of case-based reasoning to the tasks of health care planning. In: Wess S, Althoff K, Richter M (eds) Topics in case-based reasoning. Springer, Berlin, pp 365–378

Bratko I, Mozetic I, Lavac N (1989) KARDIO: a study in deep and qualitative knowledge for expert systems. MIT Press, Cambridge

Brennan A (2000) The Institute of Medicine report on medical errors-could it do harm? N Engl J Med 342(15):1123–1125

Chawla A, Gunderman R (2008) Defensive medicine: prevalence, implications, and recommendations. Acad Rad 15(7):948–949

Cheater F, Closs S (1997) The effectiveness of methods of dissemination and implementation of clinical guidelines: a selective review. Clin Eff Nurs 1:4–15

Clark D (1990) Numerical and symbolic approaches to uncertainty management in AI. Artif Intell Rev 4(2):109–146

Codish S, Shiffman R (2005) A model of ambiguity and vagueness in clinical practice guideline recommendations. AMIA Annu Symp Proc 2005:146–150

De Clercq P, Blom J, Korsten H, Hasman A (2004) Approaches for creating computer-interpretable guidelines that facilitate decision support. Artif Intell Med 31(1):1–27

Dempster A (1967) Upper and lower probability inferences based on a sample from a finite univariate population. Biometrika 54(3–4):515

Dennis S, Edwards S, Partridge M, Pinnock H, Qureshi S (2004) The dissemination of the British guideline on the management of asthma. Respir Med 98:832–837

Dinh H, Lee C, Niyato D, Wang P (2011) A survey of mobile cloud computing: architecture, applications, and approaches. Wirel Commun Mob Comput. doi:10.1002/wcm.1203/pdf

Dolin R, Alschuler L (2011) Approaching semantic interoperability in Health Level Seven. J Am Med Inform Assoc 18(1):99–103

Elkin P, Peleg M, Lacson R, Bernstam E, Tu S, Boxwala A (2000) Toward standardization of electronic guideline representation. MD Comput 17(6):39–44

Fox J, Thomson R (1998) Decision support and disease management: a logic engineering approach. IEEE Trans Inf Technol B 2(4):217–228

Fox J, Johns N, Rahmanzadeh A (1998) Disseminating medical knowledge: the PROforma approach. Artif Intell Med 14(2):157–182

Fox J, Glasspool D, Bury J (2001) Quantitative and qualitative approaches to reasoning under uncertainty in medical decision making. In: Quaglini S, Barahona P, Andreassen S (eds) Artificial intelligence in medicine—Lecture notes in computer science, vol 2101. Springer, Berlin, pp 272–282

Fox J, Alabassi A, Patkar V, Rose T, Black E (2006) An ontological approach to modelling tasks and goals. Comput Biol Med 36:837–856

Fox J, Black R, Chronakis I, Dunlop R, Patkar V, South M, Thomson R (2008) From guidelines to careflows: modelling and supporting complex clinical processes. In: Teije A, Miksch S, Lucas P (eds) Computer-based medical guidelines and protocols: a primer and current trends. IOS Press, Amsterdam, pp 44–62

Gardner R, Pryor T, Warner H (1999) The HELP hospital information system: update 1998. Int J Med Inform 54(3):169–182

Gruber T (1993) A translation approach to portable ontology specifications. Knowl Acquis 5(2):199–220

Househ M, Kushniruk A, Maclure M, Carleton B, Cloutier-Fisher D (2011) The use of conferencing technologies to support drug policy group knowledge exchange processes: an action case approach. Int J Med Inform 80:251–261

Hripcsak G (1994) Writing Arden syntax medical logic modules. Comput Biol Med 24(5):331–363

Hutchings A, Raine R, Sanderson C, Black N (2006) A comparison of formal consensus methods used for developing clinical guidelines. J Health Serv Res Policy 11(4):218–24

Isern D, Moreno A (2008) Computer-based execution of clinical. Int J Med Inform 77:787–808

Isern D, Sánchez D, Moreno A (2012) Ontology-driven execution of clinical guidelines. Comput Methods Prog Biomed 107(2):122–139

Jackson P (1990) Introduction to expert systems. Addison-Wesley Longman Publishing, Boston

John R, Zhou S-M, Garibaldi J, Chiclana F (2008) Automated group decision support systems under uncertainty: trends and future research. Int J Comput Intell Res 4(4):357–371

Johnson C, Turley J (2006) The significance of cognitive modeling in building healthcare interfaces. Int J Med Inform 75:163–172

Kaiser K, Miksch S (2009) Versioning computer-interpretable guidelines: semi-automatic modeling of 'Living Guidelines' using an information extraction method. Artif Intell Med 46(1):55–66

Kalra J (2004) Medical errors: an introduction to concepts. Clin Biochem 37(12):1043–1051

Karacapilidis N, Pappis C (1997) A framework for group decision support systems: combining AI tools and OR techniques. Eur J Oper Res 103(2):373–388

Kavanagh BP (2009) The GRADE system for rating clinical guidelines. PLoS Med 6(9):5

Kim S, Haug P, Rocha R, Choi I (2008) Modeling the Arden syntax for medical decisions in XML. Int J Med Inform 77(10):650–656

Koton P (1988) Reasoning about evidence in causal explanations. In: Mitchell T, Smith R (eds) Proceedings of the national conference on artificial intelligence AAAI-88. AAAI Press, Menlo Park, pp 256–263

Logan R, Scott P (1996) Uncertainty in clinical practice: implications for quality and costs of health care. Lancet 347:595–598

Lucas P (2004) Bayesian networks in biomedicine and health-care. Artif Intell Med 30(3):201–214

Mead P (2000) Clinical guidelines: promoting clinical effectiveness or a professional minefield? J Adv Nurs 31(1):110–116

Melle W (1978) MYCIN: a knowledge-based consultation program for infectious disease diagnosis. Int J Man Mach Stud 10(3):313–322

Miller M, Kearney N (2004) Guidelines for clinical practice: development, dissemination and implementation. Int J Nurs Stud 41(7):813–821

Neves J, Ribeiro J, Pereira P, Alves V, Machado J, Abelha A, Novais P, Analide C, Santos M, Fernández-Delgado M (2012) Evolutionary intelligence in asphalt pavement modeling and quality-of-information. Prog Artif Intel 1(1):119–135

Nikolopoulos C (1997) Expert systems: introduction to first and second generation and hybrid knowledge based systems. Marcel Dekker Inc, New York

Novais P, Salazar M, Ribeiro J, Analide C, Neves J (2010) Decision making and quality-of-information. In: Corchado E, Novais P, Analide C, Sedano J (eds) Soft computing models in industrial and environmental applications, 5th international workshop (SOCO 2010). Springer—series advances in intelligent and soft computing, vol 73, pp 187–195

Novais P, Carneiro D, Gomes M, Neves J (2012) Non-invasive estimation of stress in conflict resolution environments. In: Demazeau Y, Müller J, Corchado J, Bajo J (eds) Advances on practical applications of agents and multi-agent systems—10th international conference on practical applications of agents and multi-agent systems (PAAMS 2012) Springer—series advances in intelligent and soft computing, vol 155, pp 153–160

Oliveira T, Costa A, Novais P, Neves J (2012) An interpretable guideline model to handle incomplete information. In: Omatu S, Paz Santana J, González S, Jose M, Bernardos A, Corchado J (eds) Computing distributed, intelligence artificial, 9th international conference (DCAI 2012). Springer—series advances in intelligent and, soft computing, pp 437–444

Ohno-Machado L, Gennari J, Murphy S, Jain N, Tu S, Oliver D et al (1998) The guideline interchange format. J Am Med Inform Assoc 5(4):357–372

Ollenschläger G, Marshall C, Qureshi S, Rosenbrand K, Burgers J et al (2004) Improving the quality of health care: using international collaboration to inform guideline programmes by founding the Guidelines International Network (GIN). Qual Saf Health Care 13(6):455–460

Pearl J (1986) Fusion, propagation, and structuring in belief networks. Artif Intell 29(3):241–288

Peleg M, Wang D (2006) The guideline interchange format. Open clinical. http://www.openclinical.org/gmm_glif.html. Accessed 23 May 2012

Peleg M, Tu S, Bury J, Ciccarese P, Fox J, Greenes RA, Hall R, Johnson PD, Jones N, Kumar A, Miksch S, Quaglini S, Seyfang A, Shortliffe EH, Stefanelli M (2003) Comparing computer-interpretable guideline models: a case-study approach. J Am Med Inform Assoc 10(1):52–68

Perner P (1999) An architecture for a CBR image segmentationsystem. Eng Appl Artif Intell 12(6):749–59

Quaglini S, Stefanelli M, Cavallini A, Micieli G, Facino C, Mossa C (2000) Guideline-based careflow systems. Artif Intell Med 20(1):5–22

Ram P, Berg D, Tu S, Mansfield G, Ye Q, Abarbanel R, Beard N (2004) Executing clinical practice guidelines using the SAGE execution engine. Stud Health Technol Inform 107(1):251–255

Rosenbrand K, Croonenborg J, Wittenberg J (2008) Guideline development. In: Teije A, Miksch S, Lucas P (eds) Computer-based medical guidelines and protocols: a primer and current trends. IOS Press, Amsterdam, pp 3–21

Rosenfeld R, Shiffman R (2006) Clinical practice guidelines: a manual for developing evidence-based guidelines to facilitate performance measurement and quality improvement. Otolaryngol Head Neck 135:1–28

Ruzicka M, Svatek V (2004) Mark-up based analysis of narrative guidelines with the Stepper tool. Stud Health Technol Inform 101:132–136

Samwald M, Fehre K, Bruin J, Adlassnig K-P (2012) The Arden syntax standard for clinical decision support: experiences and directions. J Biomed Inform. doi:10.1016/j.jbi.2012.02.001

Schadow G, Mead C, Walker D (2006) The HL7 reference information model under scrutiny. Stud Health Technol Inform 124:151–156

Schalkoff R (1990) Artificial intelligence: an engineering approach. McGraw-Hill, New York

Seto E, Leonard KJ, Cafazzo JA, Barnsley J, Masino C, Ross HJ (2012) Developing healthcare rule-based expert systems: case study of a heart failure telemonitoring system. Int J Med Inform 81(8):556–565

Seyfang A, Martinez-Salvador B, Serban R, Wittenberg A, Miksch S, Marcos M, Teije AT, Rosenbrand K (2007) Maintaining formal models of living guidelines efficiently. In: Bellazzi R, Abu-Hanna A, Hunter J (eds) Proceedings of the 11th conference on artificial intelligence in medicine. Springer, Berlin, pp 441–445

Shahar Y, Miksch S, Johnson P (1998) The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artif Intell Med 14(1–2):29–51

Shahar Y, Young O, Shalom E, Galperin M, Mayaffit A, Moskovitch R, Hessing A (2004) A framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools. J Biomed Inform 37(5):325–344

Shenoy P, Shafer G (1986) Propagating belief functions with local computations. IEEE Expert 1(3):43–52

Shiffman R, Karras B, Agrawal A, Chen R, Marenco L, Nath S (2000) GEM: a proposal for a more comprehensive guideline document model using XML. J Am Med Inform Assoc 5:488–98

Shortliffe E (1993) The adolescence of AI in medicine: will the field come of age in the '90s? Artif Intell Med 5(2):93–106

Sonnenberg F, Hagerty C (2006) Computer-interpretable clinical practice guidelines: where are we and where are we going ? Method Inform Med 45(1):145–158

Sordo M, Ogunyemi O, Boxwala A, Greenes R (2004) Description and status update on GELLO: a proposed standardized object-oriented expression language for clinical decision support. Stud Health Technol Inform 107(1):164–168

Sutton D, Fox J (2003) The syntax and semantics of the PROforma guideline modeling language. J Am Med Inform Assoc 10:433–443

Straszecka E (2004) Medical knowledge representation in terms of IF-THEN rules and the Dempster-Shafer theory. In: Rutwoski L, Siekmann J, Tadeusiewicz R, Zadeh L (eds) Artificial intelligence and soft computing—ICAISC 2004, vol 3070. Springer, Berlin, pp 1056–1061

Thomson P (2000) Implementing evidence-based health care: the nurse teachers' role in supporting the dissemination and implementation of SIGN clinical guidelines. Nurs Educ Today 20(3):207–217

Thomson R, Lavender M, Madhok R (1995) How to ensure that guidelines are effective. BMJ 311:237–242

Tu S, Campbell J, Glasgow J, Nyman M, McClure R, McClay J et al (2007) The SAGE guideline model: achievements and overview. J Am Med Inform Assoc 14(5):589–598 van Gerven M, Taal B, Lucas P (2008) Dynamic Bayesian networks as prognostic models for clinical patient management. J Biomed Inform 41:515–529

Vetternlein T, Mandl H, Adlassnig K-P (2010) Fuzzy Arden syntax: a fuzzy programming language for medicine. Artif Intell Med 49:1–19

Votruba P, Miksch S, Kosara R (2004) Facilitating knowledge maintenance of clinical guidelines and protocols. Stud Health Technol Inform 107(1):57–61

Wang D, Peleg M, Tu S, Boxwala A, Ogunyemi O, Zeng Q et al (2004) Design and implementation of the GLIF3 guideline execution engine. J Biomed Inform 37(5):305–318

Weijden T, Boivin A, Burgers J, Schunemann H, Elwyn G (2011) Clinical practice guidelines and patient decision aids: an inevitable relationship. J Clin Epidemiol 65(6):584–589

Woolf S, Grol R, Hutchinson A, Eccles M, Grimshaw J (1999) Potential benefits, limitations and harms of clinical guidelines. Br Med J 318(7182):527–530

Zadeh L (1975) Fuzzy logic and approximate reasoning. Synthese 30(3):407–428

Zheng K, Padman R, Johnson M, Hasan S (2010) Guideline representation ontologies for evidence—based medicine practice. In: Khoumbati K, Dwivedi Y, Srivastava A, Lal B (eds) Handbook of research on advances in health informatics and electronic healthcare applications: global adoption and impact of information communication technologies. IGI Global, New York, pp 234–254

Zhou Z, Jiang Y, Yang Y, Chen S (2002) Lung cancer cell identification based on artificial neural network ensembles. Artif Intell Med 24(1):25–36

2.2 Representation of Clinical Practice Guideline Components in OWL

Title	Representation of Clinical Practice Guideline Components in OWL
Authors	Tiago Oliveira, Paulo Novais, and José Neves
Publication Type	Conference proceedings
Chapter	Trends in Practical Applications of Agents and Multiagent Systems
Book series	Advances in Intelligent Systems and Computing
Conference	11th International Conference on Practical Applications of Agents and Multi-Agent Systems
Publisher	Springer International Publishing Switzerland
Volume	221
Pages	77-85
Year	2013
Online ISBN	978-3-319-00563-8
Print ISBN	978-3-319-00562-1
URL	http://link.springer.com/chapter/10.1007/978-3-319-00563-8_10
State	Published
Scimago journal rank (2013)	0.139, Computer Science (Q4), Control and Systems Engineering (Q4)
Indexation	Scopus, DBLP

Contribution of the doctoral candidate
The doctoral candidate, Tiago José Martins Oliveira, declares to be the main author and the major contributor of the paper *Representation of Clinical Practice Guideline Components in OWL*.

Representation of Clinical Practice Guideline Components in OWL

Tiago Oliveira, Paulo Novais, and José Neves

Abstract. The objective of clinical decision support systems is to improve the quality of care and, if possible, help to reduce the occurrence of clinical malpractice cases such as medical errors and defensive medicine. To do so they need a machine-readable support to integrate the recommendations of Clinical Practice Guidelines. CompGuide is a Computer-Interpretable Guideline model developed in Ontology Web Language that offers support for administrative information concerning a guideline, workflow procedures, and the definition of clinical and temporal constraints. When compared to other models of the same type, besides having a comprehensive task network model, it introduces new temporal representations and the possibility of reusing pre-existing knowledge and integrating it in a guideline.

Keywords: Clinical Practice Guidelines, Ontology, OWL, Clinical tasks, Decision Support.

1 Introduction

Among the healthcare community the occurrence of medical errors and defensive medicine are top concerns [1][2]. Medical errors refer to mistakes during the clinical process that may lead to adverse events, i.e., alterations in a patient's health condition for the worse [1]. They include errors of execution, treatment and planning, and their incidence rates, although not very high, are synonymous with increased spending and loss of life quality for both physicians and patients [2]. To prevent these situations from happening, healthcare professionals often adopt another type of harmful behavior, defensive medicine. Defensive medicine consists in avoiding the treatment of difficult clinical cases to prevent possible

Tiago Oliveira · Paulo Novais · José Neves
CCTC/DI, University of Minho,
Braga, Portugal
e-mail: {toliveira,pjon,jneves}@di.uminho.pt

J.B. Pérez et al. (Eds.): *Trends in Prac. Appl. of Agents & Multiagent Syst.*, AISC 221, pp. 77–85.
DOI: 10.1007/978-3-319-00563-8_10

lawsuits or ordering additional complementary means of diagnostic motivated by the sense of self-preservation of healthcare professionals. This behaviour is also motivated by the overreliance on technological means for diagnostic purposes which in turn is responsible for driving up healthcare costs [3]. If one wants to reduce the impact of medical malpractice, it is necessary to standardize healthcare delivery and provide adequate evidence-based recommendations for clinical encounters [4]. Clinical Practice Guidelines (CPGs) are the current medium of choice to disseminate evidence-based medicine.

According to the definition of the Institute of Medicine (IOM) of the United States (US), CPGs are systematically developed statements that contain recommendations for healthcare professionals and patients about appropriate medical procedures in specific clinical circumstances [5]. They are regarded by healthcare professionals as vehicles through which they can integrate the most current evidence into patient management [6]. However, some limitations are detected in the current format of CPGs. They are available as very long documents that are difficult to consult, since only a small part of these documents are actually clinical recommendations. Moreover, there are some issues concerning the ambiguity that these documents may have [7], namely: the misunderstanding of medical terms (semantic ambiguity); conflicting instructions (pragmatic ambiguity); and the incorrect structure of statements (syntactic ambiguity). Additionally, some forms of vagueness may occur in the text, mainly due to the use of temporal terms (e.g. always, sometimes), probabilistic terms (e.g. probable, unlikely) and quantitative terms (e.g., many, few). A structured format for CPGs that is, at the same time, machine-readable would help to solve these issues by providing an adequate support for guideline dissemination and deployment, at the point and moment of healthcare delivery [8].

This work presents a representation model for CPGs developed in Ontology Web Language (OWL) capable of accommodating guidelines from any category (diagnosis, evaluation, management and treatment) and medical specialty (e.g., family practice, pediatrics, cardiology). As for the organization of this article, it presents in section two the fundamentals about OWL along with some observations concerning the advantages of choosing this knowledge representation formalism over traditional ones, like relational databases. The model, which was named CompGuide, is presented in section three with the different requirements that were taken into consideration during the development phase. Section four presents a discussion about the advantages of the model in comparison with the existing ones and provides some conclusion remarks as well as future directions for this research.

2 Advantages of Ontology Web Language

The OWL Web Ontology Language is a standard developed by the World Wide Web Consortium (W3C) and its current version is OWL 2 [9]. OWL 2 is an update to OWL with increased expressive power with regards to properties, extended support for data types and database style keys. OWL is designed for use by applications that need to process the content of information rather than just presenting information to humans. This formalism facilitates machine interpretability and is built upon other technologies such as XML, RDF and RDF-schema. The advantages of this knowledge representation formalism over RDF are related with the fact that OWL, despite being based on RDF, adds more vocabulary for describing properties and classes (e.g., disjunction, transitivity, simmetry). OWL is composed of three sublanguages: OWL Lite, OWL DL and OWL Full. The sublanguage used for this work was OWL DL and it is named this way due to its correspondence with description logics. An ontology is used to describe the concepts in a domain as well as the relationships that hold between them. To accomplish this task OWL ontologies define three basic components:

- Classes: sets that contain individuals described using formal (mathematical) descriptions that state precisely the requirements for membership of the class;
- Individuals: objects of the domain and instances of classes; and
- Properties: binary relations on individuals that may be used to link two individuals (object properties) or an individual to a data element (data properties).

The advantages of OWL reside in the manner a system uses the information. Machines do not speak human language and, sometimes, there is content that escapes their grasp. For instance, a human knows that in some cases some words are definitely related, although they are not synonyms. A machine does not recognize these relationships, but semantics are important. The idea of OWL is to provide a machine with a semantic context. So the advantage is the creation of a better management of information and descriptions.

In OWL, semantic data is assembled in a graph database that is unlike the more common relational and hierarchical databases (nodes and tables). The relationships in OWL assume a greater importance and are the carriers of the semantic content of individuals. Moreover, it is possible to describe or restrain class membership using these relations and thus accurately delimit their scope. In relational databases this would be a hard task to perform. In fact, there is several software engines developed to reason about the semantic content of ontologies, called reasoners, which check the integrity of the constraints posed on individuals in order to assert if they belong or not to a certain class. Such reasoners are Pellet, FaCT++ and HermiT which are available as plugins for Protégé, the ontology editor and knowledge acquisition system used in this work. The reasoner used in this work was FaCT++. As the objec-tive is the development of a standard machine-readable representation of CPGs, OWL seems to be most suitable.

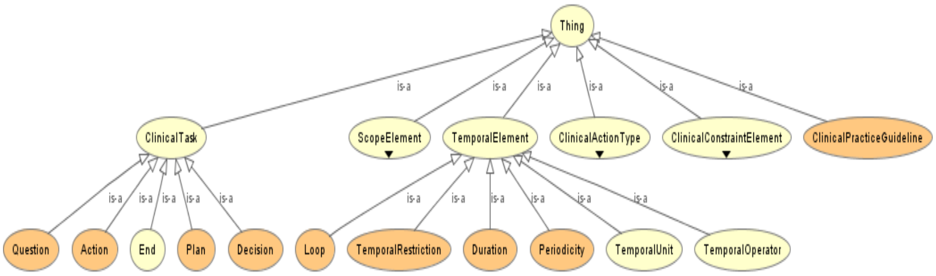


Fig. 1 Diagram of the main primitive classes of the model for Computer-Interpretable Guidelines

3 CompGuide Ontology

There are essentially two ways of developing Computer-Interpretable Guidelines (CIGs): by consulting domain experts in order to get the representation primitives or by researching different CPGs and determine the information needs of clinical recommendations. The method followed in this work was a hybrid one in the sense that it included opinions from healthcare professionals and the observation of guidelines collected from the National Guideline Clearinghouse (NGC).

The main primitive classes of the model are depicted in Fig. 1 and will be described in detail in the following subsections.

3.1 Representation of Administrative Information

As it may be seen in Fig. 1, a CPG is represented as an instance from the *ClinicalPracticeGuideline* class. To keep track of different guideline versions and to provide rigorous descriptions of guideline content and objectives, the individuals of this class have a set of data properties that represent administrative information.

OWL has built-in data types that allow the expression of simple text, numeric values and dates. As such the *string* and *date-time* properties defined for administrative purposes were: *Authorship*, *guidelineName*, *guidelineDescription*, *DateOfCreation*, *DateOfLastUpdate* and *VersionNumber*. There are also additional properties that specify in which conditions and to whom the CPG should be applied, such as *ClinicalSpecialty*, *GuidelineCategory*, *intendedUsers* and *targetPopulation*.

3.2 Construction of Workflow Procedures

CPGs are, essentially, clinical recommendations that are usually presented as sets of tasks that must be performed during clinical encounters, and/or disease management processes. To represent these tasks, CompGuide proposes three main primitive classes, defined under *ClinicalTask*: *Plan*, *Action*, *Decision* and *Question*. The tasks of an individual from *ClinicalPracticeGuideline* are all contained in an individual from *Plan*, to which it is linked through the *hasPlan* object property, as it may be seen in Fig. 2. This is an excerpt of the Standards of Medical Care in Diabetes guideline from the American Diabetes Association, extracted from the NGC. A *Plan* contains any number of instances of the other tasks, including other *Plans*, and it is connected to its first task through the *hasFirstTask* property. In turn, this task is linked to the following task in the workflow by the *nextTask* property and so on. This assures the definition of a sequence of tasks in a manner similar to a linked list, like it is shown in Fig. 2.

The remaining task classes represent different types of activities. Starting with the *Action* class, it represents a step performed by a healthcare agent that includes clinical procedures, clinical exams, medication recommendations and non-medication recommendations. The *hasClinicalActionType* object property connects an *Action* to the different action types defined in *ClinicalActionType*, with appropriate data properties to describe each one.

To express decision moments in the workflow, there is a *Decision* class. The use of this class entails a bifurcation in the clinical workflow and a choice between two or more options. The association of a *Decision* with options and rules is done through object properties that connect them to instances from the *ClinicalConstraintElement* subclasses. The next task in the clinical workflow is selected according to the outcome of the *Decision*. As so, the connection between these tasks is done using the *alternativeTasks* property. This assures that a task is executed instead of another as the result of an inference process guided by trigger conditions.

On the other hand, there may be cases when some tasks must be executed simultaneously, like the procedures of a treatment plan that act synergistically to produce a certain result. For these cases, CompGuide provides the *parallelTasks* property.

The *Question* class is used to obtain information about a patient's health condition, more specifically about the clinical parameters necessary to follow the guideline. In order to fulfill this requirement, there are data properties created to specify the name of the parameter to be obtained and the units in which it should be expressed. Such properties are *string* data types named *Parameter* and *Unit*, respectively.

The *End* class is used to signal the termination of the execution thread that is being followed and to indicate that the guideline reached its finishing point.

3.3 Definition of Temporal Constraints

The importance of time in clinical observations is paramount [10]. When assessing a patient state, a healthcare professional must take into consideration for how long the patient is manifesting his symptoms and try to fit this knowledge in the one he already has about possible causes and solutions. The recommendations of CPGs contain specifications about their temporal execution, namely intended duration, number of repetitions and cycles.

To represent all the temporal constraints, CompGuide provides the *TemporalElement* class. This class contains the two main temporal constructs, *Duration* and *Loop*. The *Duration* class specifies how long a task should last and is defined exclusively for *Plans* and *Actions*. It has a *double* data type property called *DurationValue* where a value for the intended duration of either *Actions* or *Plans* is provided. The *TemporalUnit* class, also defined under *TemporalElement*, contains individuals that represent the different time units in which the *Duration* can be expressed, namely *second*, *minute*, *hour*, *day*, *week*, *month* and *year*. In the *Loop* class it is possible to define cycles for the executions of certain tasks (*Plans* and *Actions*). Each instance of *Loop* has a data property called *RepetitionValue* which is an integer that expresses the number of repetitions that a group of tasks is subjected to. Moreover, each instance also has a *hasPeriodicity* object property that connects it to individuals from the *Periodicity* class (another subclass of *TemporalElement)* which has the appropriate constructs, namely the *hasTemporalUnit* object property and the *PeriodicityValue* data property, to define the regular intervals at which the task is repeated.

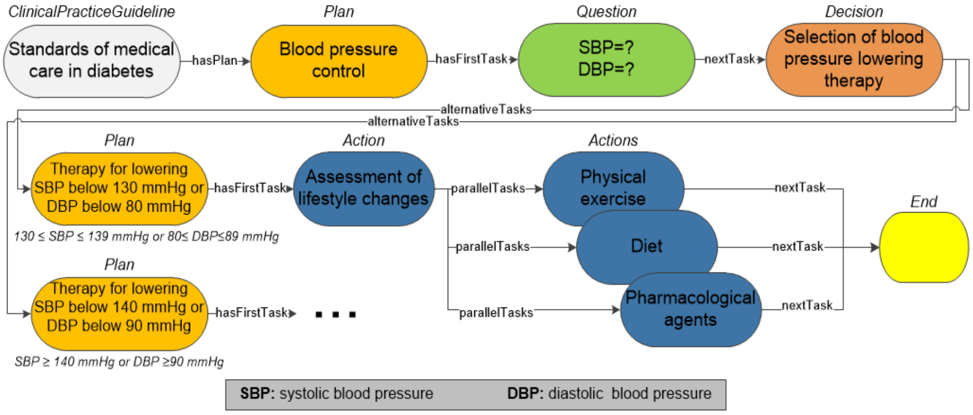

ClinicalPracticeGuideline
Standards of medical care in diabetes
hasPlan
Plan
Blood pressure control
hasFirstTask
Question
SBP=? DBP=?
nextTask
Decision
Selection of blood pressure lowering therapy
alternativeTasks
alternativeTasks
Plan
Therapy for lowering SBP below 130 mmHg or DBP below 80 mmHg
hasFirstTask
Action
Assessment of lifestyle changes
130 ≤ SBP ≤ 139 mmHg or 80≤ DBP≤89 mmHg
parallelTasks
parallelTasks
parallelTasks
Actions
Physical exercise
Diet
Pharmacological agents
nextTask
nextTask
nextTask
End
Plan
Therapy for lowering SBP below 140 mmHg or DBP below 90 mmHg
hasFirstTask
▪ ▪ ▪
SBP ≥ 140 mmHg or DBP ≥90 mmHg
SBP: systolic blood pressure
DBP: diastolic blood pressure

Fig. 2 Excerpt of the Standards of Medical Care in Diabetes clinical guideline from the American Diabetes Association represented according to the CompGuide model

Another feature of the temporal properties is the possibility to define temporal restrictions in clinical constraints. For this purpose one associates a *TemporalRestriction* and a *TemporalOperator* to clinical conditions that must be met for a task to be executed. The temporal operators are based on the theory for quality checking of clinical guidelines by Peter Lucas [11] and include the following individuals:

- *Somewhere in the past*: the condition manifested itself at some point in the past;
- *Always in the past*: the condition manifested itself continuously during a time interval in the past; and
- *Currently*: the condition is manifesting itself during the medical observations.

The *TemporalRestriction* bounds the *TemporalOperator* to a certain period of time. It possesses a *double* data property called *temporalRestrictionValue* and the *hasTemporalUnit* object property. For instance, if an *Action* requires the verification if a patient has been doing is medication correctly for the last 3 months, then *TemporalOperator* is set to *always in the past*, *temporalRestrictionValue* is set to *3* and finally *TemporalUnit* is set to *month*.

3.4 Definition of Clinical Constraints

As it was mentioned previously, a *Decision* implies the choice between two and more options. The association of individuals from *Option* to *Decision* is done by the *hasOption* property. The number of times this property is used in a *Decision* is equal to the number of options the task presents. Each *Option* has a *Parameter* data property and a *NumericalValue* or *QualitativeValue* data properties. The rules that dictate the option selection are provided by the *hasConditionSet* property, linking the individuals from this task to *ConditionSet*. The last one gathers all the necessary conditions through *hasCondition*. In *Condition*, it is possible to define the clinical parameter whose value will be compared, the unit it should be in and the operator that should be used. The *hasComparisonOperator* property connects individuals from *Condition* to *ComparisonOperator*. There, the following individuals were created: *equal to*, *greater than*, *greater or equal than*, *less than*, *less or equal than* and *different from*.

After a medical decision, it is necessary to select the next task in the clinical workflow. Therefore there must be some kind of reference in the tasks that are up for selection (connected by the *alternativeTasks* property) to the possible results of the *Decision*. This is done through the *TriggerCondition* class that also uses the *ConditionSet*. The execution of an activity is triggered when the conditions match the decision output, the selected option. There other classes in *ClinicalConstraintElement* that also use *ConditionSet* in a similar way, namely *PreCondition* and *Outcome*. *PreCondition* is used for all types of tasks to express the requirements of the patient state hat must be met before the execution of a task. For instance, when administering some pharmacological agent it should be known that the patient is not allergic to it. The *Outcome* class puts a restriction to *Plans* and *Actions* that are oriented by therapy goals, like the case of Fig. 2 in which the *Plans* will only be considered completed when the desired levels of SBP and DBP are achieved.

4 Discussion and Conclusions

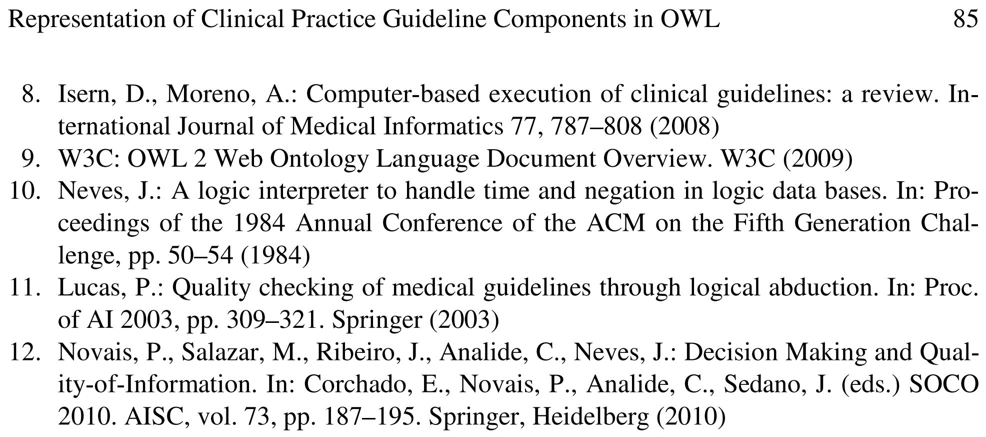
Although this work draws some inspiration from pre-existing models [8], such as Arden Syntax, PROforma, GLIF3, Asbru or SAGE, it also introduces different views about the definition of clinical constraints, temporal properties, clinical task scheduling and how all these aspects connect with each other. Taking as a reference the oldest, and probably, the most widely (academically) used model, Arden Syntax (now a standard of Health Level 7), which represents knowledge for only one clinical decision, CompGuide provides more expressive power by allowing the definition of a clinical workflow, similarly to GLIF3 and PROforma. However, these models do not have native methods for expressing temporal constraints, using a subset of Asbru temporal language to deal with this issue. Asbru, is, by far, the model that has more temporal constructors and the most complete in this regard, but, at same time, is considered very complex and, in some cases, impractical. The temporal constructs presented in this work are intended to be a compromise between expressivity and complexity that better suits the necessities of clinical decision support systems and thus of healthcare professionals. Another important aspect is the possibility of reusing knowledge from other ontologies in CompGuide by merging the two. This way, the scalability of knowledge is assured.

None of the current formalisms for CPGs is used in a large scale for real context clinical decision support systems. Yet, there is evidence that CIG based decision support could in fact improve the quality of care and address the previously mentioned problems [12]. By using OWL to represent CPGs, one intends to benefit from the advantages of this knowledge representation formalism and if possible, increase the penetration of CIGs into routine medical care.

Acknowledgments. This work is funded by national funds through the FCT – Fundação para a Ciência e a Tecnolo-gia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011". The work of Tiago Oliveira is supported by a doctoral grant by FCT (SFRH/BD/85291/2012).

References

1. Kalra, J.: Medical errors: an introduction to concepts. Clinical Biochemistry 37, 1043–1051 (2004)
2. Chawla, A., Gunderman, R.B.: Defensive medicine: prevalence, implications, and recommendations. Academic Radiology 15, 948–949 (2008)
3. Hermer, L.D., Brody, H.: Defensive medicine, cost containment, and reform. Journal of General Internal Medicine 25, 470–473 (2010)
4. Landrigan, C.P., Parry, G.J., Bones, C.B., Hackbarth, A.D., Goldmann, D.A., Sharek, P.J.: Temporal trends in rates of patient harm resulting from medical care. The New England Journal of Medicine 363, 2124–2134 (2010)
5. Field, M.J., Lohr, K.N. (eds.): Committee on Clinical Practice Guidelines, I. of M. Guidelines for Clinical Practice:From Development to Use. The National Academies Press (1992)
6. Vachhrajani, S., Kulkarni, A.V., Kestle, J.R.W.: Clinical practice guidelines. Journal of Neurosurgery: Pediatrics 3, 249–256 (2009)
7. Codish, S., Shiffman, R.N.: A Model of Ambiguity and Vagueness in Clinical Practice Guideline Recommendations. In: AMIA Annual Symposium Proceedings, pp. 146–150 (2005)



8. Isern, D., Moreno, A.: Computer-based execution of clinical guidelines: a review. International Journal of Medical Informatics 77, 787–808 (2008)
9. W3C: OWL 2 Web Ontology Language Document Overview. W3C (2009)
10. Neves, J.: A logic interpreter to handle time and negation in logic data bases. In: Proceedings of the 1984 Annual Conference of the ACM on the Fifth Generation Challenge, pp. 50–54 (1984)
11. Lucas, P.: Quality checking of medical guidelines through logical abduction. In: Proc. of AI 2003, pp. 309–321. Springer (2003)
12. Novais, P., Salazar, M., Ribeiro, J., Analide, C., Neves, J.: Decision Making and Quality-of-Information. In: Corchado, E., Novais, P., Analide, C., Sedano, J. (eds.) SOCO 2010. AISC, vol. 73, pp. 187–195. Springer, Heidelberg (2010)

2.3 ASSESSING AN ONTOLOGY FOR THE REPRESENTATION OF CLINICAL PROTOCOLS IN DECISION SUPPORT SYSTEMS

Title	Assessing an Ontology for the Representation of Clinical Protocols in Decision Support Systems
Authors	Tiago Oliveira, Paulo Novais, and José Neves
Publication Type	Conference proceedings
Chapter	Trends in Practical Applications of Agents and Multiagent Systems
Series	Advances in Intelligent Systems and Computing
Conference	13th International Conference on Practical Applications of Agents and Multi-Agent Systems
Publisher	Springer International Publishing Switzerland
Volume	372
Pages	47-54
Year	2015
Online ISBN	978-3-319-19629-9
Print ISBN	978-3-319-19628-2
URL	http://link.springer.com/chapter/10.1007/978-3-319-19629-9_6
State	Published
Scimago journal rank (2015)	0.153, Computer Science (Q4), Control and Systems Engineering (Q4)
Indexation	ISI Web of Science, Scopus, DBLP

Contribution of the doctoral candidate
The doctoral candidate, Tiago José Martins Oliveira, declares to be the main author and the major contributor of the paper *Assessing an Ontology for the Representation of Clinical Protocols in Decision Support Systems*.

Assessing an Ontology for the Representation of Clinical Protocols in Decision Support Systems

Tiago Oliveira, Paulo Novais, and José Neves

Algoritmi Research Centre/Department of Informatics,
University of Minho, Braga, Portugal
{toliveira,pjon,jneves}@di.uminho.pt

Abstract. In order to assess the expressiveness of the CompGuide ontology for Clinical Practice Guidelines, a study was conducted with fourteen students of the Integrated Masters in Biomedical Engineering from the University of Minho in Portugal to whom it was proposed the representation of multiple guidelines according to the ontology. They were then asked to evaluate the ontology through a questionnaire and written reports. Although the results seem promising, there is the need for significant improvements mainly in: the representation of medication prescriptions, the tasks used to retrieve information from the patient, the diversity of actions offered by the ontology, the expressiveness of conditions regarding the state of a patient, and temporal constraints.

Keywords: Clinical Protocols, Ontologies, Clinical Decision Support.

1 Introduction

There are various ways of expressing medical knowledge in Clinical Decision Support Systems (CDSSs) [5], but, among them, decision trees, probabilistic models, and task-network models (TNM) are arguably the most popular [8]. Despite the obvious advantages of each one, task-network models are still preferred over the others because of their representation of clinical guideline knowledge in hierarchical structures containing networks of clinical actions that unfold over time. The main TNMs include Asbru [9], PROforma [2], GLIF3 [1], SAGE [12], and GLARE [11], among others. The first one, Asbru [9], focuses on temporal parameters and offers constructs to define starting points, durations, and ending points of tasks. In addition, it allows the specification of intentions for actions and prescriptions, as well as the expected outcomes. These time-oriented actions, conditions and intentions are represented as patterns which assume the hierarchical structure of plans and sub-plans. As for PROforma [2], it follows a structure somewhat similar to Asbru in the sense that it also resorts to the representation of guidelines as plans. There is a root task to which every plan in a guideline belongs.In turn, a plan has any number of instances of other tasks, from actions to decisions. Its focus is on argumentation in favor or against a decision. GLIF3 [1] was the first model to place its emphasis on the use of standards. In addition to using the task model just as the other models do, it makes use

J. Bajo et al. (eds.), *Trends in Prac. Appl. of Agents, Multi-Agent Sys. and Sustainability*,
Advances in Intelligent Systems and Computing 372, DOI: 10.1007/978-3-319-19629-9_6

of terminologies to avoid semantic ambiguity in the definition of clinical terms, and employs the HL7 Reference Information Model (RIM) to ensure that other systems can communicate with a system using GLIF3. As for SAGE [12], a direct evolution of GLIF3, it is considered one of the most complete approaches to Computer-Interpretable Guidelines (CIGs). This model places a high importance on the notion of context. The context coordinates the activation of guideline-based decision support. As such, it has constructs that allow the definition of the conditions in which medical practice takes place, whether they are related to the organizational setting and roles, the patient characteristics, or necessary resources. The procedural guideline logic is represented in an activity graph constrained by the already-mentioned context variables. The GuideLine Acquisition, Representation and Execution (GLARE) [11] model is specialized in the treatment of repeated (periodic) events, which play a major role in clinical therapies. There are other important models featured in comprehensive reviews such as [8] and [6]. However, there is no standard computer-interpretable representation for CPGs, and many of the existing are criticized for lack of expressiveness.

Ontologies are explicit representations of the concepts from a domain. They are the basic construction units of the semantic web and their objective is to allow applications to process the content of information, rather than just presenting it. The Web Ontology Language (OWL) [3] is a standard proposed by the World Wide Web Consortium (W3C) designed for facilitating machine interpretation. Ontologies have not been widely explored in the representation of the procedural logic in clinical protocols, yet, they provide an ideal support for this knowledge. Thus, the objective is exploring OWL as the underlying language for CIGs and use it to develop a CIG ontology with the intention of building a sufficiently expressive representation that would be capable of accommodating knowledge from different types of guidelines.

This work presents a preliminary study designed to assess the CompGuide ontology for clinical protocols. As such, the organization of the paper is as follows. Section two provides a brief description of the ontology with its main primitive classes and properties. Section three describes the materials and methods for the study. Section four presents the results and their discussion. Finally, in section five conclusions and future work considerations are provided.

2 Developed Ontology

The CompGuide ontology was initially presented in [7]. The ontology provides a task network model representation for clinical guidelines in OWL. In order to fulfill that purpose, it follows a logic in which complex information elements are represented as individuals with multiple object properties connecting them to other individuals, and simple information that cannot be further decomposed is represented using data properties. However, simple information that is reusable and will most likely be needed across different guidelines is represented as class individuals as well. In this regard the representation is similar to a linked list of procedures. As such, a CPG is represented as an instance of the *ClinicalPracticeGuideline* class. Individuals from this class have a set of data and object properties that enable the representation of descriptive and administrative guideline information such as the name of the guideline, its general description, date of creation and last update, version, clinical specialty, category, intended users, and target population. An example of the initial definition of a guideline is given in Fig. 1.

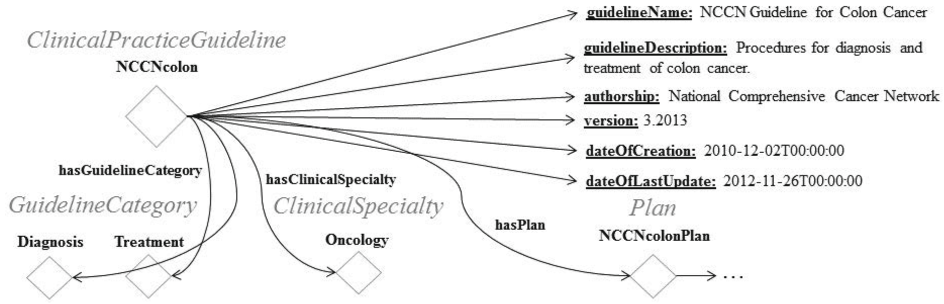

ClinicalPracticeGuideline
NCCNcolon
guidelineName: NCCN Guideline for Colon Cancer
guidelineDescription: Procedures for diagnosis and treatment of colon cancer.
authorship: National Comprehensive Cancer Network
version: 3.2013
dateOfCreation: 2010-12-02T00:00:00
dateOfLastUpdate: 2012-11-26T00:00:00
hasGuidelineCategory
GuidelineCategory
Diagnosis
Treatment
hasClinicalSpecialty
ClinicalSpecialty
Oncology
hasPlan
Plan
NCCNcolonPlan
...

Fig. 1. Initial definition of a National Comprehensive Cancer Network guideline for the treatment of colon cancer in the CompGuide ontology.

Every guideline is connected to an individual of the class *Plan* which, in turn, is connected to other individuals that represent basic tasks. The procedural logic and workflow of clinical tasks is represented using three basic classes: *Action*, *Decision* and *Question*. The objective here is to create a recommendation plan that contains references to specific types os tasks. The *Action* class expresses a procedure that should be carried out by a health care professional. There are several subtypes of actions in the ontology that specify their nature with more detail. The *Decision* class is used to make assertions about the state of the patient, to infer new information from the existing one. The most obvious example of such a task is clinical diagnosis. The *Question* task is used to get information about the symptoms of a patient, to register information from the observations of the physician, and to store results from clinical exams. This type of task gathers all the information necessary for the execution of the clinical algorithm. Through object properties, it is possible to define the sequence of execution of tasks or if they should be executed simultaneously or concurrently.

3 Materials and Methods

The objective of the preliminary study was to assess the expressivity of the CompGuide ontology in four important aspects of CPGs, namely the representation of administrative information, the construction of workflow procedures, the definition of temporal constraints, and the definition of clinical constraints.These are considered the fundamental aspects of CIG representation and the pre-requisites of a good CIG model [8]. For that purpose, 14 students from the fourth year of the Integrated Masters in Biomedical Engineering, branch in Medical Informatics, from the University of Minho, in Braga, Portugal, aged between 22 and 23 years old, were selected. They had no prior knowlegde of OWL, and received training in both OWL and Protégé [10] for a total of six hours distributed by three two-hour sessions. After the training, they were taught about the structure, classes, and properties of CompGuide in a two-hour session.

Then, the students were asked to do an assignment which consisted in the representation of a CPG in the referred ontology using Protégé. They were handed a .owl file containing the definition of the ontology which they should fill in by adding the necessary elements. The set of CPGs used in the assignment is showed in Table 1. They were randomly distributed among the students. As much as possible, one tried that each guideline included multiple categories, namely diagnosis, evaluation, treatment, and management. The assignment had the duration of one month, by the end of which the students were asked to fill in a questionnaire which consisted of sixteen statements regarding the expressiveness of the model in the four above-mentioned aspects. The statements used in the questionnaire complete the general statement: "The CompGuide ontology allowed the representation of:". Statements 1-9 were about the construction of workflow procedures, statements 10-12 were related with the definition of clinical constraints, statements 13-15 were devised to assess the definition of temporal constraints, and, finally, statement 16 was about the representation of administrative information. The set of statements can be consulted in Figure 2. The answers were provided in a five point Likert rating scale [4] (*1-strongly disagree*, *2-disagree*, *3-neutral*, *4-agree*, *5-strongly agree*). It was also asked that the students handed a ten-page report describing their principal difficulties and observations while performing the task.

Table 1. List of the guidelines that were used in the study, featuring their name, organization and the number of people assigned to their representation.

Clinical Practice Guideline	Organization	People Assigned
Clinical Practice Guidelines in Oncology - Colon Cancer	National Comprehensive Cancer Network	2
Clinical Practice Guidelines in Oncology - Rectal Cancer	National Comprehensive Cancer Network	2
Clinical Pratice Guidelines in Oncology - Distress	National Comprehensive Cancer Network	2
Clinical Practice Guidelines in Oncology - Palliative Care	National Comprehensive Cancer Network	2
Detection,Evaluation,and Treatment of High Blood Cholesterol in Adults	National Heart Lung and Blood Institute	1
Diagnosing and Managing Asthma	National Heart Lung and Blood Institute	1
Diagnosis, Evaluation and Management of von Willebrand Disease	National Heart Lung and Blood Institute	1
Diagnosis and Treatment of Respiratory Illness in Children and Adults	Institute for Clinical Systems Improvement	1
Diagnosis and Management of Diabetes	Institute for Clinical Systems Improvement	1
Diagnosis and Treatment of Ischemic Stroke	Institute for Clinical Systems Improvement	1

The process resulted in the diverging stacked bar chart in Figure 2. The chart presents the total percentage of agreement (calculated as *agree + strongly agree*), the total percentage of disagreement (calculated as *disagree + strongly disagree*), and the percentage of participants who were neutral (equal to the percentage of the *neutral* category), for each statement, in order to show the central tendency in each item.

4 Results and Discussion

By consulting the chart of Figure 2, and specifically items 1 to 9 which refer to the representation of different procedures and tasks in a workflow, it is possible to verify that, for each item in this group, the level of agreement is at least equal or above 50%. Indeed, the item about medication prescriptions (item 1) is the one that has the lowest agreement, the highest percentage in the *neutral* category (43%), and the only one that has percentage in the *strongly disagree* category (7%). This is indicative that the representation of medication prescriptions may have issues. In fact, in the reports the participants mentioned that the representation of medication prescriptions was impractical at times. In the ontology, a medication has to be defined as the subtype of an *Action* individual, and one *Action* can only have one prescription. However, in several of the represented guidelines what were perceived as single actions included the administration of more than one drug, requiring the representation of medication schemes as several parallel actions instead of a single action with a clear objective. Another criticism to the representation of medication prescriptions was that it was mandatory to define an active ingredient, dosage, pharmaceutical form, and posology for a drug, but in certain guidelines these elements were not

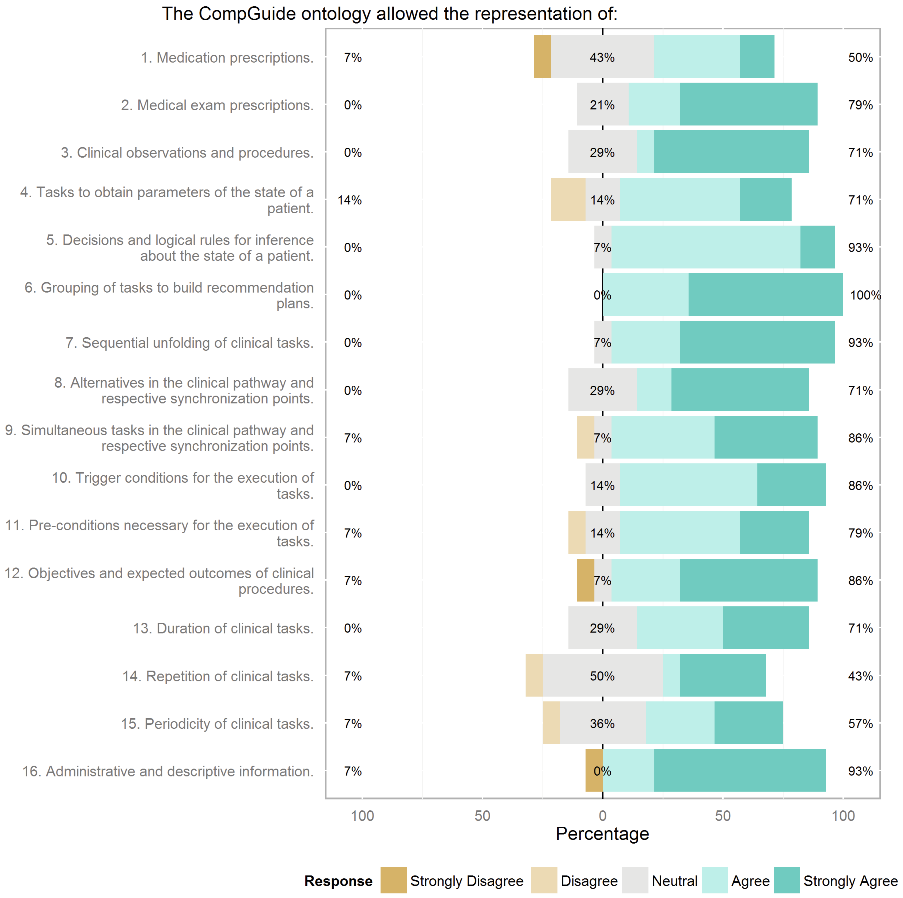
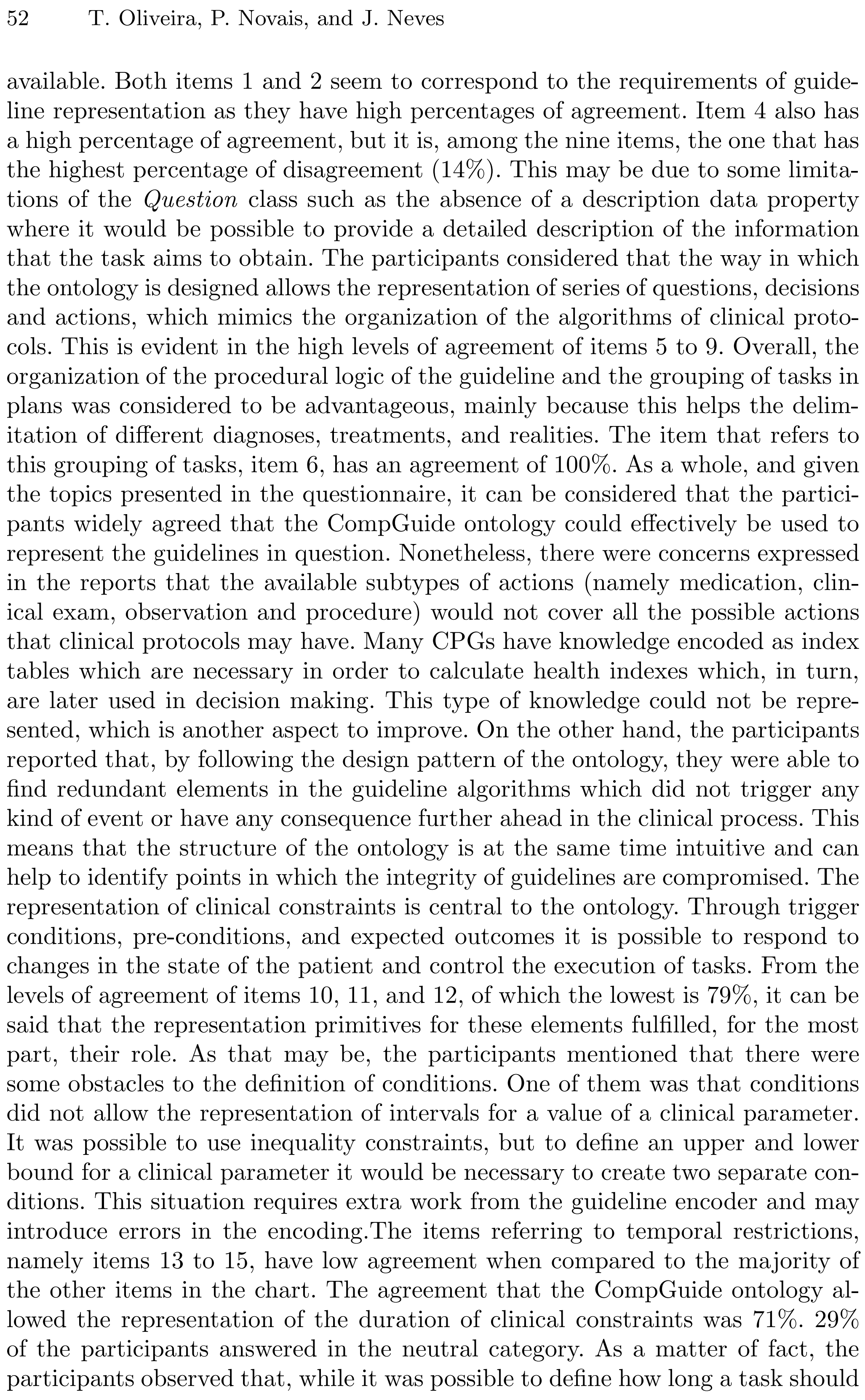


Fig. 2. Diverging stacked bar chart showing the results of the questionnaire to assess the expressiveness of the CompGuide ontology



available. Both items 1 and 2 seem to correspond to the requirements of guideline representation as they have high percentages of agreement. Item 4 also has a high percentage of agreement, but it is, among the nine items, the one that has the highest percentage of disagreement (14%). This may be due to some limitations of the *Question* class such as the absence of a description data property where it would be possible to provide a detailed description of the information that the task aims to obtain. The participants considered that the way in which the ontology is designed allows the representation of series of questions, decisions and actions, which mimics the organization of the algorithms of clinical protocols. This is evident in the high levels of agreement of items 5 to 9. Overall, the organization of the procedural logic of the guideline and the grouping of tasks in plans was considered to be advantageous, mainly because this helps the delimitation of different diagnoses, treatments, and realities. The item that refers to this grouping of tasks, item 6, has an agreement of 100%. As a whole, and given the topics presented in the questionnaire, it can be considered that the participants widely agreed that the CompGuide ontology could effectively be used to represent the guidelines in question. Nonetheless, there were concerns expressed in the reports that the available subtypes of actions (namely medication, clinical exam, observation and procedure) would not cover all the possible actions that clinical protocols may have. Many CPGs have knowledge encoded as index tables which are necessary in order to calculate health indexes which, in turn, are later used in decision making. This type of knowledge could not be represented, which is another aspect to improve. On the other hand, the participants reported that, by following the design pattern of the ontology, they were able to find redundant elements in the guideline algorithms which did not trigger any kind of event or have any consequence further ahead in the clinical process. This means that the structure of the ontology is at the same time intuitive and can help to identify points in which the integrity of guidelines are compromised. The representation of clinical constraints is central to the ontology. Through trigger conditions, pre-conditions, and expected outcomes it is possible to respond to changes in the state of the patient and control the execution of tasks. From the levels of agreement of items 10, 11, and 12, of which the lowest is 79%, it can be said that the representation primitives for these elements fulfilled, for the most part, their role. As that may be, the participants mentioned that there were some obstacles to the definition of conditions. One of them was that conditions did not allow the representation of intervals for a value of a clinical parameter. It was possible to use inequality constraints, but to define an upper and lower bound for a clinical parameter it would be necessary to create two separate conditions. This situation requires extra work from the guideline encoder and may introduce errors in the encoding.The items referring to temporal restrictions, namely items 13 to 15, have low agreement when compared to the majority of the other items in the chart. The agreement that the CompGuide ontology allowed the representation of the duration of clinical constraints was 71%. 29% of the participants answered in the neutral category. As a matter of fact, the participants observed that, while it was possible to define how long a task should last, the expressive power of the ontology was limited in this regard. It was not possible to define intervals of duration for tasks with minimum and maximum values. However, this type of information element occurred very often in the guidelines. Meanwhile, item 14, which concerns the repetition of clinical tasks, gathered only 43% of agreement, and 50% of the participants answered in the *neutral* category. Despite recognizing the usefulness of the ontology element that enables the definition of the number of times that a task should be executed, the participants believed that a crucial element was missing, and that was conditional repetitions, i.e., the possibility of stating that a task should be repeated if a the state of a patient does not improve. This was also an observation made within the scope of item 15.Finally, item 16 got 93% of agreement, which seems to convey that the ontology elements responsible for the representation of administrative information such as authoring, name of the guideline, its general description, date of creation, and so forth, are fulfilling their target function.

On a final note, representation formats such as CompGuide have to also be capable of representing situations in which the decision is left to the health care professional. Occasionally, the elements for a decision may not be all present in the description provided by a CPG, and in such situations guidelines may recommend that health care professionals follow their best judgment according to the available evidence.

5 Conclusions and Future Work

Although there was no access to an entire statistical population of interest, given the time-consuming nature of the survey, the study still provides useful hints for the development of the CompGuide ontology. Essentially, one may consider that the guidelines used in the survey were accurately represented according to the ontology, despite the need for certain adaptations, which did not affect the logic of the clinical process represented in CPGs. Nonetheless, there is a need for significant improvements, mainly in: the representation of medication prescriptions, the tasks used to retrieve information from the patient, the diversity of actions offered by the ontology, the expressiveness of conditions regarding the state of a patient, and temporal constraints as a whole.

These promising results may be due, in part, to the nature of the sample used in the survey. Since all of the participants were students of medical informatics, one can say that they have a clearer understanding than most about the role played by technology as a support for medical knowledge, and, although they were not familiar with OWL or Protégé, they already knew about similar models and principles. Moreover, the study should have included a broader range of CPGs in terms of origin and clinical specialty in order to expose the participants to a wider diversity of clinical situations. In future surveys, and once the issues identified are addressed, these aspects should be taken into consideration. Future work also includes completing the ontology proposed in this work with the international standards and data models proposed by HL7.

Acknowledgements. This work is part-funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-028980 (PTDC/EEI-SII/1386/2012). The work of Tiago Oliveira is supported a FCT grant with the reference SFRH/BD/85291/ 2012.

References

1. Boxwala, A.A., Peleg, M., Tu, S., Ogunyemi, O., Zeng, Q.T., Wang, D., Patel, V.L., Greenes, R.A., Shortliffe, E.H.: GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. Journal of Biomedical Informatics 37(3), 147–161 (2004)
2. Fox, J., Ma, R.T.: Decision Support for Health Care: the PROforma Evidence Base. Informatics in Primary Care 14(1), 49–54 (2006)
3. Group, O.W.: OWL 2 Web Ontology Language Document Overview. Tech. Rep. W3C (October 2009), `http://www.w3.org/TR/owl2-overview/`
4. Jamieson, S.: Likert scales: how to (ab)use them. Medical Education 38(12), 1217–1218 (2004), `http://dx.doi.org/10.1111/j.1365-2929.2004.02012.x`
5. Musen, M., Shahar, Y., Shortliffe, E.: Clinical decision-support systems. In: Shortliffe, E., Cimino, J. (eds.) Biomedical Informatics. Health Informatics, pp. 698–736. Springer, Heidelberg (2006)
6. Oliveira, T., Novais, P.: Development and implementation of clinical guidelines: An artificial intelligence perspective. Artificial Intelligence Review 42(4), 999–1027 (2014)
7. Oliveira, T., Novais, P., Neves, J.: Representation of Clinical Practice Guideline Components in OWL. In: Pérez, J.B., Hermoso, R., Moreno, M.N., Rodríguez, J.M.C., Hirsch, B., Mathieu, P., Campbell, A., Suarez-Figueroa, M.C., Ortega, A., Adam, E., Navarro, E. (eds.) Trends in Pract. Appl. of Agents & Multiagent Syst. AISC, vol. 221, pp. 77–85. Springer, Heidelberg (2013)
8. Peleg, M.: Computer-interpretable clinical guidelines: A methodological review. Journal of Biomedical Informatics 46(4), 744–763 (2013)
9. Seyfang, A., Miksch, S., Marcos, M.: Combining diagnosis and treatment using ASBRU. International Journal of Medical Informatics 68(1-3), 49–57 (2002)
10. Stanford Center for Biomedical Informatics Research: Protege Wiki (2014), `http://protegewiki.stanford.edu/wiki/Main_Page`
11. Terenziani, P., Montani, S., Bottrighi, A., Torchio, M., Molino, G., Correndo, G.: The GLARE approach to clinical guidelines: main features. Studies in Health Technology and Informatics 101(3), 162–166 (2004)
12. Tu, S.W., Campbell, J.R., Glasgow, J., Nyman, M.A., McClure, R., McClay, J., Parker, C., Hrabak, K.M., Berg, D., Weida, T.: Others: The SAGE Guideline Model: achievements and overview. Journal of the American Medical Informatics Association 14(5), 589 (2007)

2.4 DECISION SUPPORT PROVIDED BY A TEMPORALLY ORIENTED HEALTH CARE ASSISTANT

Title	Decision Support Provided by a Temporally Oriented Health Care Assistant: An Implementation of Computer-Interpretable Guidelines
Authors	Tiago Oliveira, António Silva, José Neves, and Paulo Novais
Publication Type	Journal
Publication Name	Journal of Medical Systems
Publisher	Springer US
Volume	41
Number	1
Pages	1–13
Year	2016
Month	November
Online ISSN	1573-689X
Print ISSN	0148-5598
URL	http://link.springer.com/article/10.1007/s10916-016-0655-6
State	Published online
Scimago journal rank (2015 - last known)	0.72, Health Informatics (Q2), Health Information Management (Q1), Information Systems (Q2), Medicine (Q2)
JCR impact factor (2014 - last known)	2.213, Health Care Sciences and Services (Q2), Medical Informatics (Q2)

Contribution of the doctoral candidate
The doctoral candidate, Tiago José Martins Oliveira, declares to be the main author and the major contributor of the paper *Decision Support Provided by a Temporally Oriented Health Care Assistant: An Implementation of Computer-Interpretable Guidelines*.

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

EDUCATION & TRAINING

Decision Support Provided by a Temporally Oriented Health Care Assistant

An Implementation of Computer-Interpretable Guidelines

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

Received: 31 May 2016 / Accepted: 31 October 2016

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

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

✉ Tiago Oliveira
toliveira@di.uminho.pt

António Silva
pg25307@alunos.uminho.pt

José Neves
jneves@di.uminho.pt

Paulo Novais
pjon@di.uminho.pt

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

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

Introduction

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

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

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

Published online: 26 November 2016

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

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

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

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

Related work

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

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

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

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

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

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

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

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

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

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

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

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

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

It is possible to see that each model has at least one limitation in one type of temporal constraint. From that, one may conclude that the efforts in defining new CIG models have not been coupled with a simultaneous effort to devise comprehensive temporal constructs in said models.

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

Development of the health care assistant

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

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

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

Architecture of the supporting system

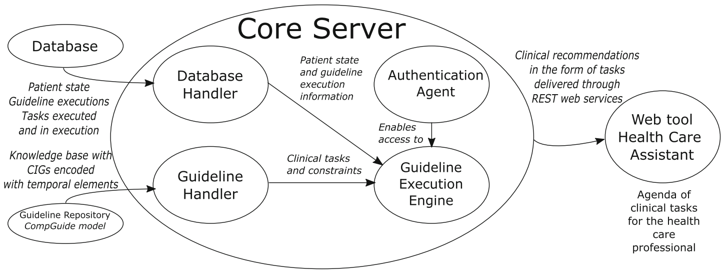
The architecture of the CompGuide system is shown in Fig. 1. Its main component is a *Core Server* that encapsulates the most important modules of the system. The *Core Server* provides all the required services to allow external applications, such as web applications or mobile applications, to execute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

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

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

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

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



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

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

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

Representation of computer-interpretable guidelines

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

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

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

Temporal elements of guidelines

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

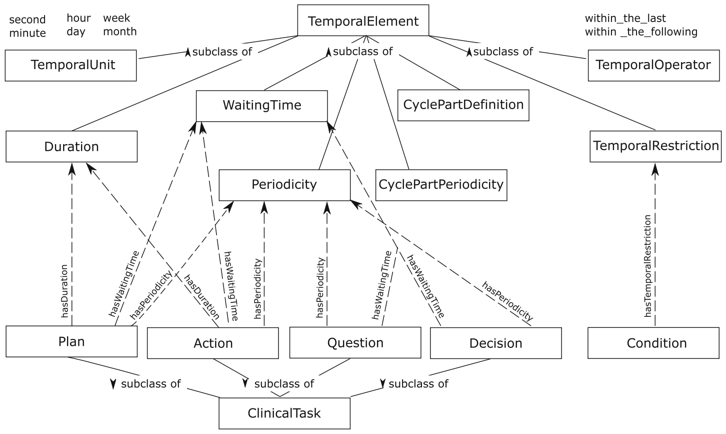


Fig. 2 Classes of the temporal model used in CompGuide

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

Temporal constraints about the execution of clinical tasks

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

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

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

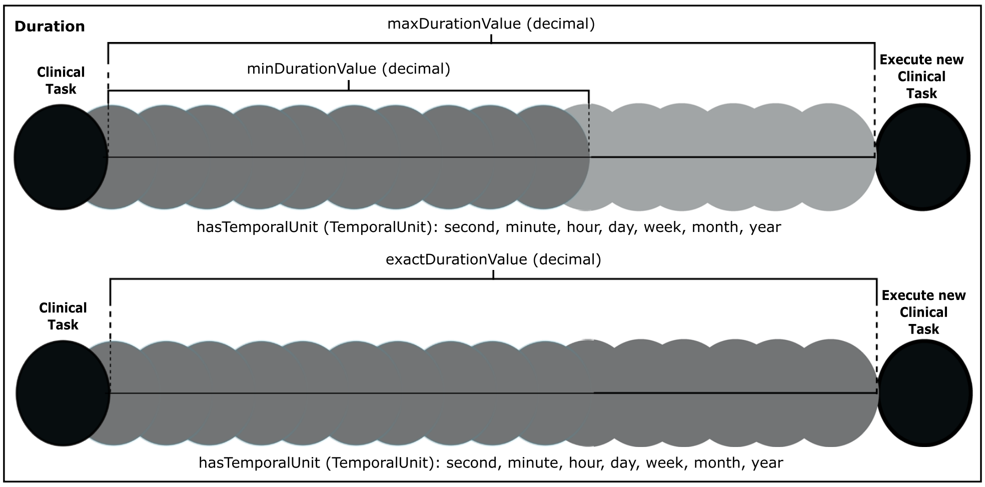

Duration
maxDurationValue (decimal)
minDurationValue (decimal)
Clinical Task
Execute new Clinical Task
hasTemporalUnit (TemporalUnit): second, minute, hour, day, week, month, year
exactDurationValue (decimal)
Clinical Task
Execute new Clinical Task
hasTemporalUnit (TemporalUnit): second, minute, hour, day, week, month, year

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

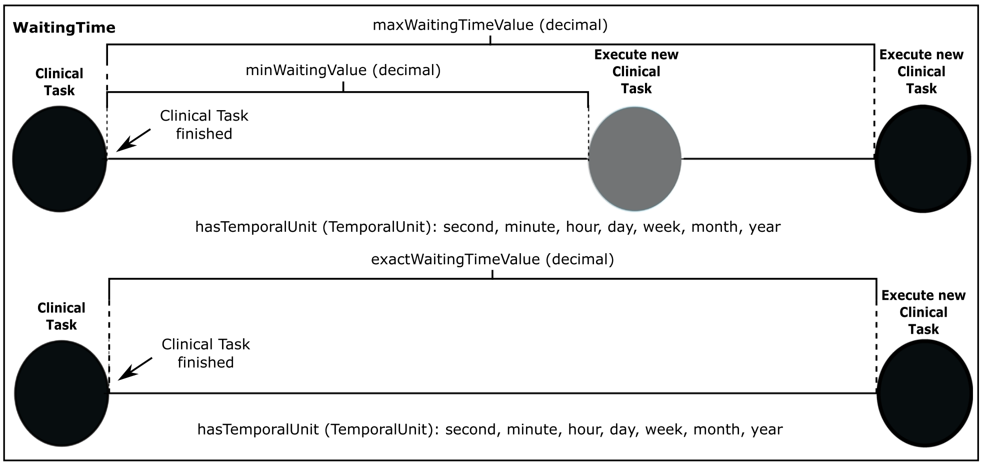

WaitingTime
maxWaitingTimeValue (decimal)
minWaitingValue (decimal)
Clinical Task
Execute new Clinical Task
Execute new Clinical Task
Clinical Task finished
hasTemporalUnit (TemporalUnit): second, minute, hour, day, week, month, year
exactWaitingTimeValue (decimal)
Clinical Task
Execute new Clinical Task
Clinical Task finished
hasTemporalUnit (TemporalUnit): second, minute, hour, day, week, month, year

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

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

Temporal constraints about the state of a patient

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

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

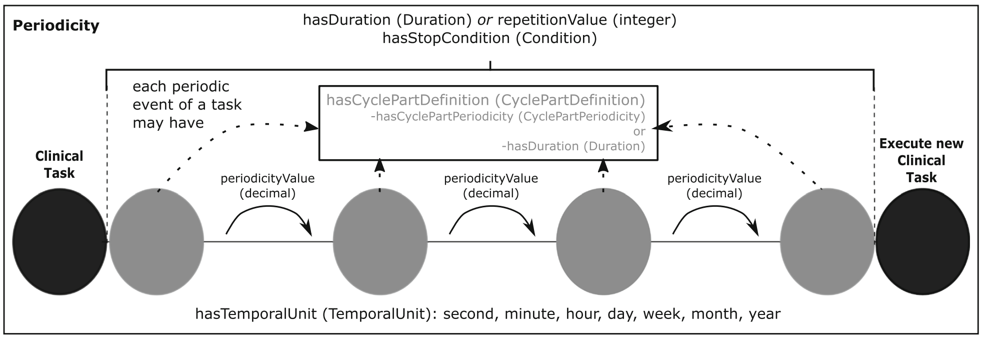

Periodicity
hasDuration (Duration) or repetitionValue (integer)
hasStopCondition (Condition)
each periodic event of a task may have
hasCyclePartDefinition (CyclePartDefinition)
-hasCyclePartPeriodicity (CyclePartPeriodicity)
or
-hasDuration (Duration)
Clinical Task
Execute new Clinical Task
periodicityValue (decimal)
periodicityValue (decimal)
periodicityValue (decimal)
hasTemporalUnit (TemporalUnit): second, minute, hour, day, week, month, year

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

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

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

Case-study featuring a guideline for colon cancer treatment

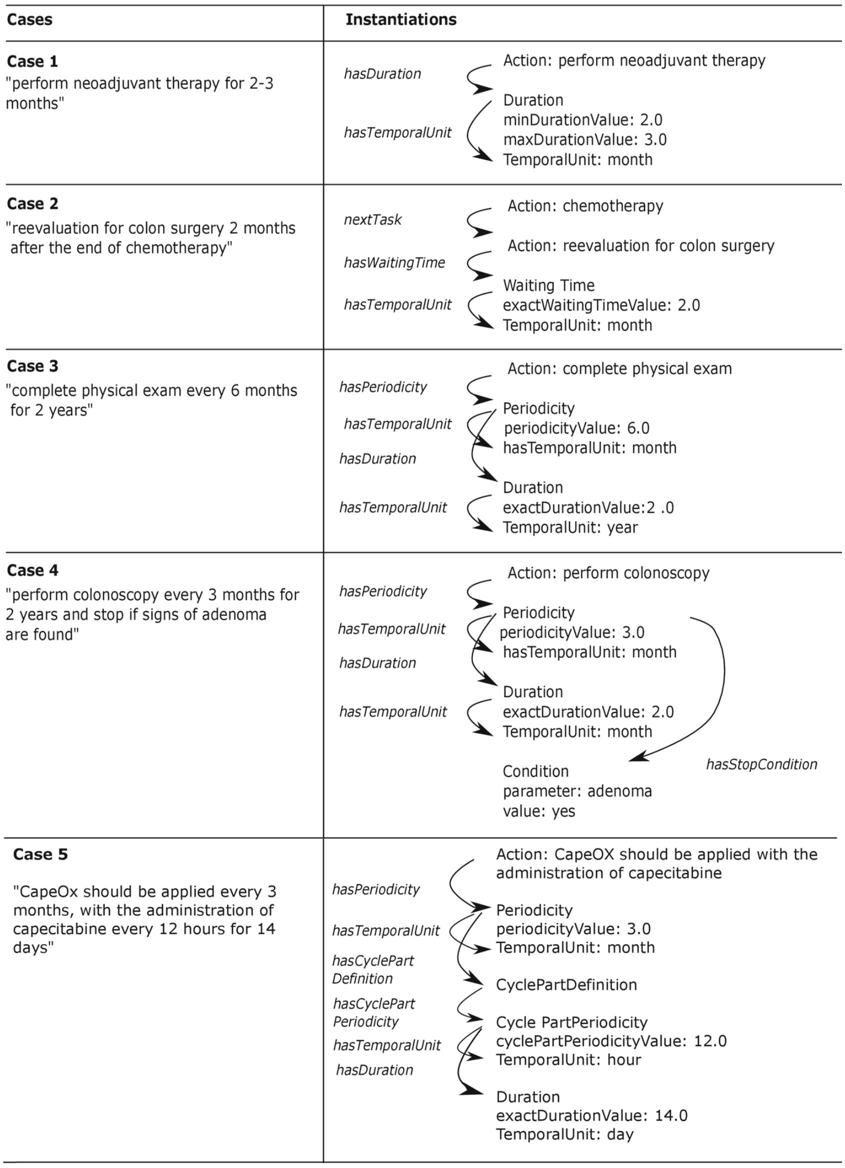
In order to verify the expressiveness of the temporal representation model, a guideline of the National Comprehensive Cancer Network (NCCN) [3] was fully represented using the CompGuide ontology. The CPG is used for the diagnosis and management of colon cancer and, thus, contains many and varied clinical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

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

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

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

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



Cases	Instantiations
Case 1 "perform neoadjuvant therapy for 2-3 months"	Action: perform neoadjuvant therapy *hasDuration* Duration minDurationValue: 2.0 maxDurationValue: 3.0 *hasTemporalUnit* TemporalUnit: month
Case 2 "reevaluation for colon surgery 2 months after the end of chemotherapy"	Action: chemotherapy *nextTask* Action: reevaluation for colon surgery *hasWaitingTime* Waiting Time exactWaitingTimeValue: 2.0 *hasTemporalUnit* TemporalUnit: month
Case 3 "complete physical exam every 6 months for 2 years"	Action: complete physical exam *hasPeriodicity* Periodicity periodicityValue: 6.0 *hasTemporalUnit* hasTemporalUnit: month *hasDuration* Duration exactDurationValue:2 .0 *hasTemporalUnit* TemporalUnit: year
Case 4 "perform colonoscopy every 3 months for 2 years and stop if signs of adenoma are found"	Action: perform colonoscopy *hasPeriodicity* Periodicity periodicityValue: 3.0 *hasTemporalUnit* hasTemporalUnit: month *hasDuration* Duration exactDurationValue: 2.0 *hasTemporalUnit* TemporalUnit: month *hasStopCondition* Condition parameter: adenoma value: yes
Case 5 "CapeOx should be applied every 3 months, with the administration of capecitabine every 12 hours for 14 days"	Action: CapeOX should be applied with the administration of capecitabine *hasPeriodicity* Periodicity periodicityValue: 3.0 *hasTemporalUnit* TemporalUnit: month *hasCyclePart Definition* CyclePartDefinition *hasCyclePart Periodicity* Cycle PartPeriodicity cyclePartPeriodicityValue: 12.0 *hasTemporalUnit* TemporalUnit: hour *hasDuration* Duration exactDurationValue: 14.0 TemporalUnit: day

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

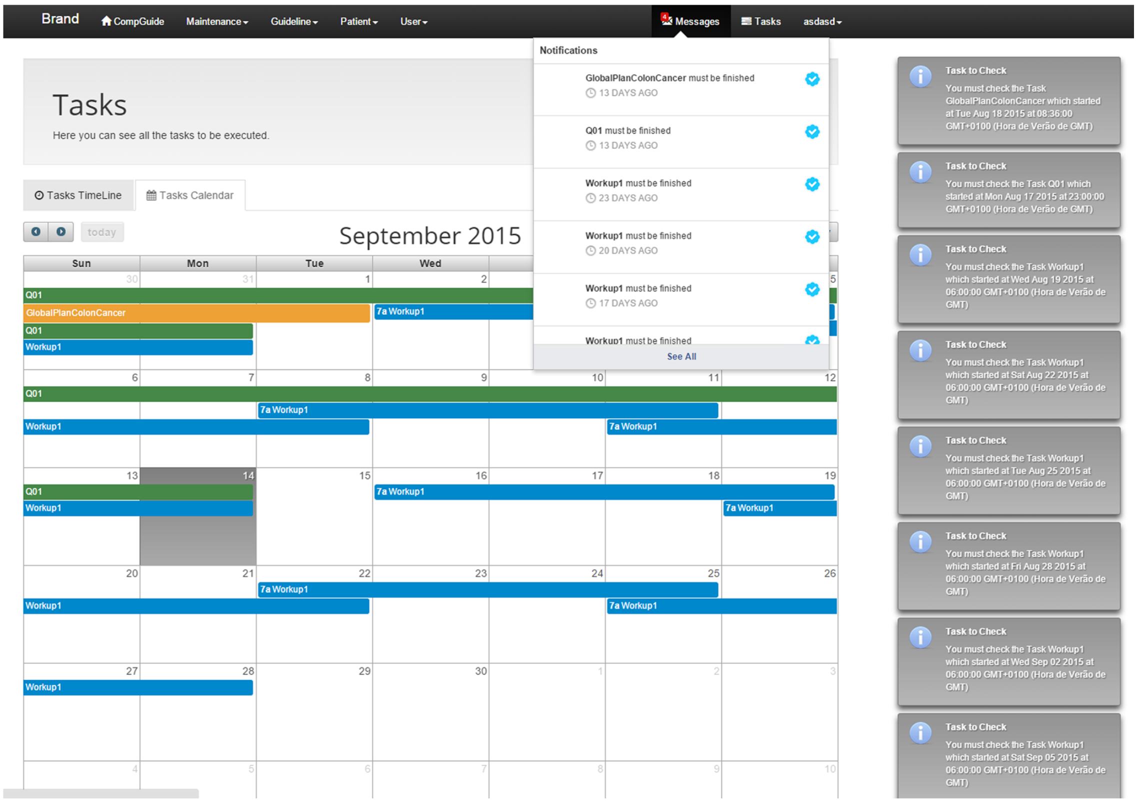
of periodicities is the most important aspect in view of the frequency with which this temporal pattern appears in CPGs. In this regard, the handling of nested periodicities is similar to the one performed in GLARE. An important difference in interpretation is that of repetition conditions. In the proposed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

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

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

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

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



Notifications

GlobalPlanColonCancer must be finished
13 DAYS AGO

Q01 must be finished
13 DAYS AGO

Workup1 must be finished
23 DAYS AGO

Workup1 must be finished
20 DAYS AGO

Workup1 must be finished
17 DAYS AGO

Workup1 must be finished

Task to Check
You must check the Task GlobalPlanColonCancer which started at Tue Aug 18 2015 at 08:36:00 GMT+0100 (Hora de Verão de GMT)

Task to Check
You must check the Task Q01 which started at Mon Aug 17 2015 at 23:00:00 GMT+0100 (Hora de Verão de GMT)

Task to Check
You must check the Task Workup1 which started at Wed Aug 19 2015 at 06:00:00 GMT+0100 (Hora de Verão de GMT)

Task to Check
You must check the Task Workup1 which started at Sat Aug 22 2015 at 06:00:00 GMT+0100 (Hora de Verão de GMT)

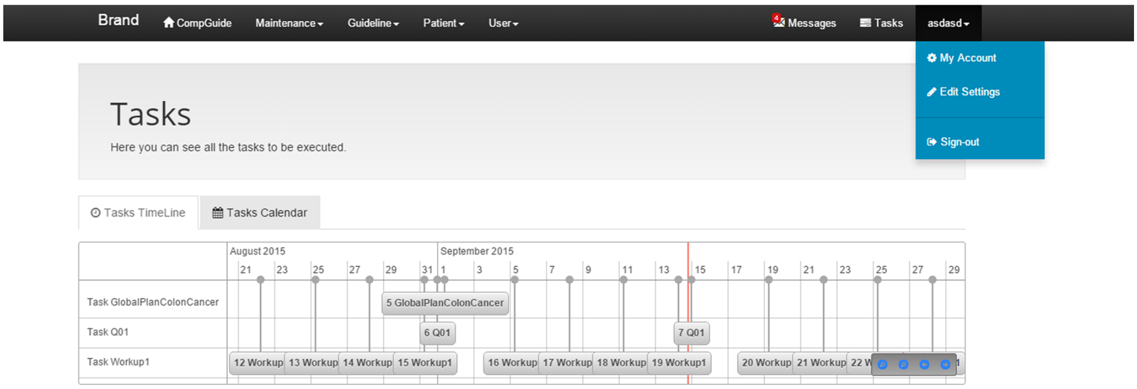
Task to Check
You must check the Task Workup1 which started at Tue Aug 25 2015 at 06:00:00 GMT+0100 (Hora de Verão de GMT)

Task to Check
You must check the Task Workup1 which started at Fri Aug 28 2015 at 06:00:00 GMT+0100 (Hora de Verão de GMT)

Task to Check
You must check the Task Workup1 which started at Wed Sep 02 2015 at 06:00:00 GMT+0100 (Hora de Verão de GMT)

Task to Check
You must check the Task Workup1 which started at Sat Sep 05 2015 at 06:00:00 GMT+0100 (Hora de Verão de GMT)

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



Tasks

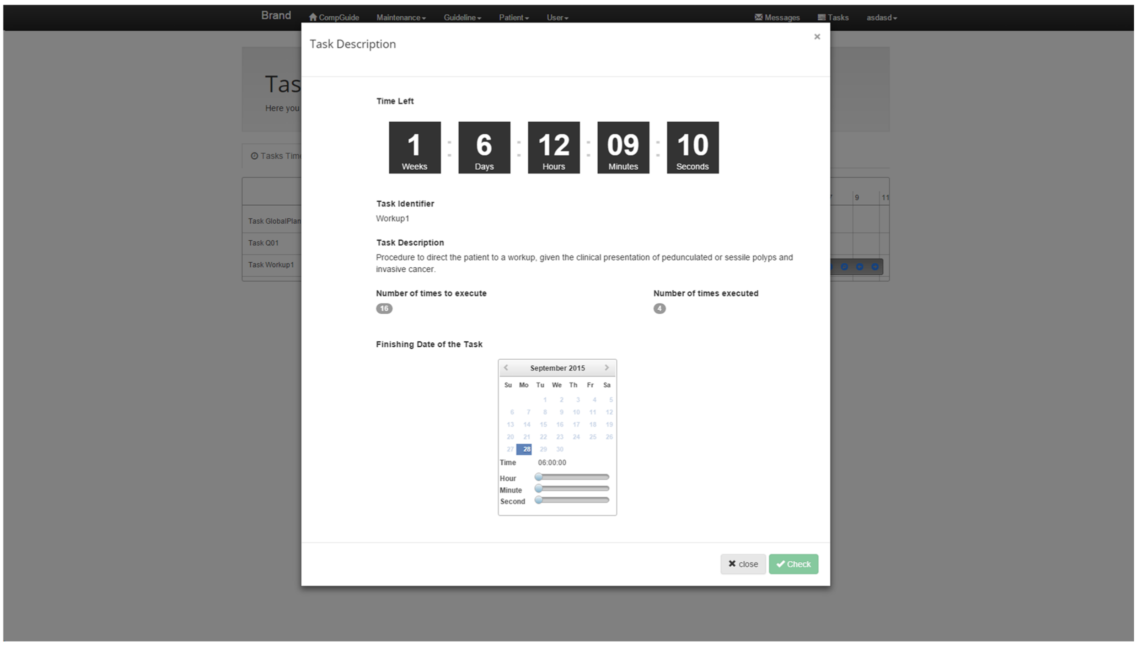
Here you can see all the tasks to be executed.

Fig. 9 Timeline task view of the Health Care Assistant

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

Web-based tool for the visualization and execution of guidelines

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



Task Description

Time Left

1 Weeks : 6 Days : 12 Hours : 09 Minutes : 10 Seconds

Task Identifier

Workup1

Task Description

Procedure to direct the patient to a workup, given the clinical presentation of pedunculated or sessile polyps and invasive cancer.

Number of times to execute: 16

Number of times executed: 4

Finishing Date of the Task

September 2015

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Time 06:00:00

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

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

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

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

Conclusions and future work

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

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

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

References

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

Informatics, Health Informatics, pp. 698–736. New York: Springer, 2006. doi:10.1007/0-387-36278-920.

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

2.5 A DYNAMIC DEFAULT REVISION MECHANISM FOR SPECULATIVE COMPUTATION

Title	A dynamic default revision mechanism for speculative computation
Authors	Tiago Oliveira, Ken Satoh, Paulo Novais, José Neves, and Hiroshi Hosobe
Publication Type	Journal
Publication Name	Autonomous Agents and Multi-Agent Systems
Publisher	Springer US
Pages	1–40
Year	2016
Month	July
Online ISSN	1573-7454
Print ISSN	1387-2532
URL	http://link.springer.com/article/10.1007/s10458-016-9341-9
State	Published online
Scimago journal rank (2015 - last known)	0.85, Artificial Intelligence (Q2)
JCR impact factor (2015 - last known)	1.417, Computer Science (Q2), Artificial Intelligence (Q2), Automation and Control Systems (Q3)

Contribution of the doctoral candidate
The doctoral candidate, Tiago José Martins Oliveira, declares to be the main author and the major contributor of the paper *A dynamic default revision mechanism for speculative computation*.

Auton Agent Multi-Agent Syst
DOI 10.1007/s10458-016-9341-9

A dynamic default revision mechanism for speculative computation

Tiago Oliveira[1] · Ken Satoh[2] · Paulo Novais[1] · José Neves[1] · Hiroshi Hosobe[3]

Abstract In this work a default revision mechanism is introduced into speculative computation to manage incomplete information. The default revision is supported by a method for the generation of default constraints based on Bayesian networks. The method enables the generation of an initial set of defaults which is used to produce the most likely scenarios during the computation, represented by active processes. As facts arrive, the Bayesian network is used to derive new defaults. The objective with such a new dynamic mechanism is to keep the active processes coherent with arrived facts. This is achieved by changing the initial set of default constraints during the reasoning process in speculative computation. A practical example in clinical decision support is described.

Keywords Default revision · Incomplete information · Speculative computation · Bayesian networks

✉ Tiago Oliveira
toliveira@di.uminho.pt

Ken Satoh
ksatoh@nii.ac.jp

Paulo Novais
pjon@di.uminho.pt

José Neves
jneves@di.uminho.pt

Hiroshi Hosobe
hosobe@hosei.ac.jp

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

[2] National Institute of Informatics, Sokendai University, Tokyo, Japan

[3] Department of Digital Media, Hosei University, Tokyo, Japan

Published online: 26 July 2016

1 Introduction

In order to tackle situations of problem solving and decision making in which cases of incomplete information may occur, a framework of speculative computation was proposed, first for yes/no questions [23,24] and then for general questions using constraints [9,22]. The term Speculative Computation is used to describe a series of procedures by which an answer for a problem is searched and computed using default beliefs. A default belief is defined here as something thought to be the case in replacement of the true information, if the last is not available. Speculative computation is formalized in terms of a framework and process semantics featuring mechanisms for the reduction and revision of processes. Such a framework allows an agent to reason with default beliefs while waiting for the other agents to reply. In this way, the computation of answers is brought forward and, if the arriving information matches the default beliefs, it is possible to save time in the search for an answer. This kind of framework is particularly useful in multi-agent systems since it is often difficult to guarantee efficient and reliable communications between agents. A multi-agent system may be deployed in an unreliable network or need human intervention to promote interactions between agents, which may cause situations that delay or even prevent communications. An agent which is part of such a system and uses information provided by the other agents may see its inference process blocked or delayed. The use of the term information here refers to the elements that serve as premises in the inference process and which will later enable the agent to derive logical conclusions, in order to solve a problem or make a decision.

In [9,22–24], fixed default beliefs are used in speculative computation. However, when applied to a real setting, the default beliefs are highly dependent on the context. The same is to say they depend on the set of circumstances and facts that surround a problem and change over time. A default belief may change according to the information obtained by later replies from the information sources and may become incoherent with the returned facts, which would result in tentative answers that do not represent the most likely scenarios. A scenario is a process representing an answer obtained through reductions based on default beliefs, meaning that, in the context of a problem, it represents an outline of a possible event.

To account for this drawback, the present work proposes a dynamic revision of default beliefs. It is important to state that this revision does not refer to the replacement of default beliefs with true information, but to their replacement with new default beliefs. It is performed on-line since it occurs as answers are searched and facts arrive. In a real setting, it is important to anticipate changes of state and therefore, in the present work, default beliefs are revised into new default beliefs as information arrives, which allows the prompt adjustment of scenarios. These default beliefs assume the form of default constraints. The same is to say that the information used to reduce a process involving a variable for which no real value is known (i.e., the default) is represented as a constraint in a constraint logic program (CLP). The behaviour of an agent is also specified in a CLP, where derivations are handled as alternative answers. The manipulation of such alternative answers is done with the processing of disjunctive constraints. In order to distinguish this framework from previous ones, the term Speculative Computation with Default Revision will be used. Its main contribution is twofold:

- The first contribution is a default generation method for Speculative Computation based on Bayesian networks (BNs). A BN featuring the variables used in the CLP is constructed with data of previous attempts of an agent to solve a problem. The probability distribution provided by the BN enables the derivation of default constraints before the beginning of the computation, without any information, and during the computation, with partial information. This represents a new use for BNs, given that, in this context, they assume a supportive role, feeding the CLP with default beliefs, and are not the main component in solving the problem.

- The second, and most important, is a mechanism for the dynamic revision of default constraints. The Speculative Computation framework is augmented with a default revision phase in which, upon the arrival of information and change in default beliefs, existing processes are revised in order to become consistent with the new defaults. A gradual convergence of scenarios towards the real answer of a problem is achieved with this revision mechanism.

The paper is organized in six sections. Section 2 provides related work about speculative computation and the use of default beliefs in concurrent constraint programming, belief revision, and defeasible logic. In Sect. 3, the Speculative Computation with Default Revision framework, with its elements and procedures, is explained. Section 3 also presents the formalization of an example using the framework and features a claim that shows the correctness of the procedures in the phases of Speculative Computation with Default Revision, thus demonstrating their effectiveness. The procedures for the Generation of default constraints are explained in Sect. 4. An execution trace of an example regarding a clinical decision support system is discussed in Sect. 5. The situations recreated in the example allow the observation of the convergence of scenarios towards the real answer of a problem. In Sect. 6, conclusions are presented along with future work considerations.

2 Related work

Speculative computation was first presented as a search algorithm that performs a speculative evaluation of expressions, i.e., if the value of an expression is not known, a computation may proceed until the value for that expression is needed [2]. The algorithm implements a form of parallelism in the evaluation of function arguments. The proposal was based on the principle that problems could be solved more quickly on parallel machines if some work could be started before it is known to be necessary. The underlying principle was later adopted by [23] to develop a decision making and problem solving framework using default beliefs for multi-agent systems in the presence of incomplete information. The framework was subsequently extended with iterative belief revision [24], default constraints and constraint processing [9,22], and abductive reasoning [21]. All these extensions targeted more dynamic and interactive environments, where there is a potential for failure in communication and for the creation of states of incomplete information. The strength of speculative computation fits systems which have a well-defined procedural logic and are based on rules. The framework performs the role of an interface that manages the state of the information and the tentative computations. In previously developed work [9,21–24], the set of default beliefs used is fixed, which may be a drawback because newly arrived information may have an impact on the remaining variables for which default beliefs are being used. As a result, the scenarios from the tentative computations might not represent the most likely outcomes in terms of answers. The Speculative Computation with Default Revision is the latest of a series of developments arising from the previously mentioned work.

The work developed in the field of concurrent constraint programming bears resemblances with the present work, namely the focus on distributed computing and computation with partial information. Some similarities with the work presented herein may be found in constraint programming languages such as the Andorra Kernel Language (AKL) [5] and the Oz programming language [26]. AKL is a concurrent programming language that uses guards. A guard consists of determinate goals, which only require the execution of one clause. When a goal is selected in the guard, it is executed across all the clauses, and those that succeed originate local speculative variable bindings. Afterwards, each successful variable binding is tested against the goals belonging to the body of the clause in a depth-first search. The use of the term speculative here refers to the uncertainty as to whether the variable bindings in the guard will later succeed or not, which represents a guess about what might happen or be true. As for Oz, it is based on the Oz Programming Model for concurrent programming. The procedures in the model allow the control of multiple computation spaces assigned to various agents. The synchronization of the agents is achieved through a centralized update of a constraint store. The speculative component of the model lies in the local computations performed by the agents, which take place with partial information. The speculative elements used in both AKL and Oz are meant for parallel computing, which means that all the possible computation pathways are simultaneously explored until they fail. On the other hand, the Speculative Computation with Default Revision framework channels the computation resources to the most plausible computation pathways, which, from the point of view of this work, are the ones worth exploring. This is achieved by extracting default constraints using a probabilistic model and using them to fill in incomplete information.

Concurrent constraint programming covers a wide variety of systems. A common denominator to all of them is that they may have to deal with partial information. In the timed default concurrent constraint programming system presented in [20], there is a timed default mechanism by which a default value is assigned to a variable if any other value, different from the default, has not been added to the constraint store at a given time. The distinctive feature of this constraint system is that there is a time limit for constraints to be added after which the processes are reduced using default beliefs. The default belief is also fixed and there is no mechanism by which it can be changed. Treating default beliefs in an isolated way automatically discards dependence relationships between variables.

Another approach sharing a few similarities with Speculative Computation with Default Revision is probabilistic concurrent constraint programming [7]. In this model, a probability mass function is assigned to a variable, which allows a constructor to choose a value for the variable according to its probability distribution. However, probabilistic constraint programming is used to model non-deterministic choices in systems that require components to exhibit different behaviours on different runs. Again, there is no notion of default belief, but there is a set of random variables with a dynamic behaviour, without fixed values. We aim to imprint this dynamic behaviour on default beliefs. From our perspective, they should not be fixed and should change according to newly arrived facts from the agent information sources. This is possible using a probabilistic model to derive new default constraints for the variables that are not covered by the newly arrived facts.

The LIFF [19] model devises an updating mechanism for logic programs based on tags added during the computation of explanations. It allows the addition of if-then rules by updating the existing solutions through the existing tags, thus removing the need to perform backtracking and computing a new answer from scratch. In Speculative Computation with Default Revision the same underlying idea of updating parts of the answers rather than computing new solutions is followed as well, with the difference that the updates are in the form of constraints, and default constraints can be updated to definitive constraints representing true information or to other default constraints.

Default revision has not been widely explored in non-monotonic logic. However, there are procedures described in the literature that are close to it. In [17], a semantics and a proof theory of a system for defeasible argumentation are presented. The prevalence of some arguments over others is determined by priorities placed on the rules that support them. The defaults in this case are the hierarchical relationships between rules. However, the priorities between arguments are not fixed and, instead, are defeasibly derived within the logic program. The proposal is based on the idea that fixed priorities between arguments in a real world problem are impractical. Similarly, in [6] the revision of rule priorities is considered in the legal domain. But, in it, defaults are revised to accommodate the preferred answers of participants. Although the relationship between these works and the work herein is only at a conceptual level, the underlying principle that fixed default information is unrealistic is the same. In Speculative Computation with Default Revision, default constraints are dynamically revised according to newly known information during reasoning, thus one may say that on-line default revision is performed, as opposed to off-line default revision in previous works. In these last two approaches, priorities are revised after facts are known and conclusions are drawn. The mechanism operates independently and is disconnected from the outside world. Conversely, herein a form of on-line default revision is proposed, which means that the mechanism interacts with the outside world during the reasoning process, while answers are computed, and changes its beliefs according to said interaction.

In [12] a Distributed Defeasible Speculative Reasoning (DDSR) framework is presented. It combines speculative computation with defeasible logic in order to model a multi-context system with autonomous logic-based agents. Internally, each agent has a set of default hypotheses about the beliefs of its peers and a local defeasible theory in which the default hypotheses are used to draw conclusions. The conclusions depend on the hierarchical relationship between the arguments that build them and the score of the beliefs. Such scores are derived from a reputation table featuring each agent in the environment. Like previous works in speculative computation, it features a process reduction phase, in which tentative conclusions are produced, and a fact arrival phase, in which beliefs change and processes are revised according to the replies from other agents. The defeasible logic component is used to sort out conflicts within the local theory of an agent. In this sense, this work covers an aspect that is not present in our proposal, which is conflicting information. However, Speculative Computation with Default Revision is focused solely on the mechanisms by which speculative computation takes place and it tackles different problems. In [12] there is no indication of how to produce the default hypotheses. Furthermore, these default hypotheses are fixed and can only be changed into their real value according to the replies from other agents. These aspects are addressed in Speculative Computation with Default Revision with the introduction of a mechanism for the generation of defaults and a new phase for the revision of default beliefs.

3 A framework for speculative computation with default revision

Speculative Computation with Default Revision has two components. The first is a Framework for Speculative Computation with Default Revision (SF_{DR}), which will be described in this section. It is where all the necessary elements to solve a problem and the different procedures to manage information are encoded. The other component is the Generation of default constraints, described in Sect. 4, which contains a BN that generates default constraints, both before and during the execution of procedures in the framework, based on the replies received by the speculative agent. The term speculative agent will be used to refer to the agent performing speculative computation. Within the setting of the multi-agent system defined for this work, this agent is responsible for providing an answer to a problem based on its internal logic theory and the information he receives from the other agents. The speculative agent hosts the SF_{DR}.

A description of the elements and procedures in the SF_{DR} is presented in the sections below. A summary of the abbreviations and symbols used throughout this section is provided in Table 1 of Annex 1.

3.1 Elements of the framework

An SF_{DR} featuring disjunctive constraint processing is defined in terms of the tuple $\langle \Sigma, \mathcal{E}, \Delta, \mathcal{P} \rangle$. This formulation is based on the work presented in [9]. The framework is hosted by a speculative agent in a multi-agent system. It is used to structure the inference process and to manage the information the agent needs from other agents in the system, in order to solve a problem. The subsequent elements of the tuple have the following meaning:

- Σ is a finite set of constants. An element in Σ is the identifier of an agent in a multi-agent system and represents an information source from which the speculative agent obtains information;
- $\mathcal{E}$ is a set of predicates called external predicates. When Q is an atom with an external predicate and S is the identifier of a remote agent information source belonging to Σ, $Q@S$ is called an askable atom;
- Δ is the default answer set and consists of a set of default constraint rules called default rules w.r.t. $Q@S$, of the following form:

  $$\text{“}Q@S \hookleftarrow C \parallel.\text{”}$$

  where

  - $Q@S$ is an askable atom;
  - C is a set of constraints called default constraints for $Q@S$;
  - A default rule w.r.t. $Q@S$ is denoted as $\delta(Q@S)$.

- $\mathcal{P}$ is a constraint logic program of the following form:

  $$\text{“}H \hookleftarrow C \parallel B_1, B_2, \ldots, B_n.\text{”}$$

  where

  - H is a positive ordinary literal called a head of rule R, denoted as $head(R)$;
  - C is a set of constraints called body constraints of rule R, denoted as $const(R)$, where $const(R)$ may be empty;
  - Each of “$B_1, B_2, \ldots, B_n$” is an ordinary literal or an askable literal. “$B_1, B_2, \ldots, B_n$” is referred to as the body of R, denoted as $body(R)$, where $body(R)$ may be empty.

In order to show how a problem can be formalized in the SF_{DR} and Speculative Computation with Default Revision can be applied to specific domains and systems, an example featuring a clinical decision support system is provided. Clinical decision support is a domain where cases of incomplete information frequently occur. The outline of the system is shown in Fig. 1, and its main purpose is to provide advice to health care professionals in the form of clinical tasks, based on machine-interpretable versions of clinical guidelines [27]. The structure of this system mirrors similar examples that can be found in the literature [10,14,16]. The elements of its architecture include:

- A *guideline engine*: responsible for interpreting clinical guideline instructions represented in the knowledge base against information about the state of a patient;
- A *knowledge base*: containing machine-interpretable versions of clinical guidelines represented in an ontology;
- A *local repository*: with information for other patients in previous executions of a clinical guideline;

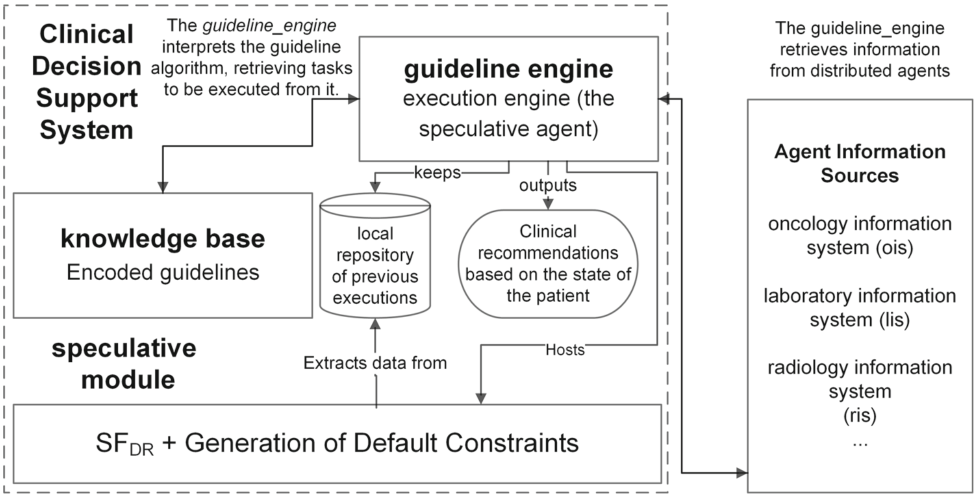

Clinical Decision Support System
The *guideline_engine* interprets the guideline algorithm, retrieving tasks to be executed from it.
guideline engine
execution engine (the speculative agent)
keeps
outputs
knowledge base
Encoded guidelines
local repository of previous executions
Clinical recommendations based on the state of the patient
speculative module
Extracts data from
Hosts
SF_{DR} + Generation of Default Constraints
The guideline_engine retrieves information from distributed agents
Agent Information Sources
oncology information system (ois)
laboratory information system (lis)
radiology information system (ris)
...

Fig. 1 Architecture of the clinical decision support system with its basic components

- A *speculative module*: an addition to the system and a module hosted by the *guideline engine* which implements an SF_{DR} with a generation of defaults mechanism.

The *guideline engine* obtains information about the state of a patient from distributed agent information sources, as described in Fig. 1, one of which is the oncology information system (*ois*). As such, the *guideline engine* is the speculative agent hosting the SF_{DR} and, together with the agent information sources, constitutes a multi-agent system.

The problem which the *guideline engine* has to solve is the choice of the clinical task with the appropriate treatment following colon cancer surgery. This decision is typically made based on the TNM Classification of Malignant Tumors, a classification in which T describes the degree of tumor invasion into the wall of the colon, N is the number of metastases in regional lymph nodes, and M represents the detection of distant metastases in other organs, such as the liver or the lungs. In cancer assessment, there are situations in which physicians are uncertain of the value of T, because tumor invasion may be difficult to assess solely from imagiology techniques. N and M may also be unknown and are dependent on the completion of laboratory exams whose results may take some time to be known. This setting was extracted from the Clinical Practice Guideline in Oncology for Colon Cancer [1], and the medical content is greatly simplified for the sake of explaining the dynamics of Speculative Computation with Default Revision.

In Fig. 2, the problem is represented according to an ontology for clinical guidelines [15] in which every step is displayed as a task. In the graph, first there is a question task, *question1*, to obtain the values of the T, N and M variables. Then, there are five action tasks, linked to the previous through the *hasAlternativeTask* property. A task is selected based on the fulfilment of trigger conditions, also depicted in Fig. 2. For instance, *action2*, which recommends the participation in a clinical trial, a period of observation, or chemotherapy with capecitabine or 5-FU/leucovorin, will only be proposed by the *guideline engine* if the value of T is $t3$ and the value of M is $m0$.

Based on this situation the example for the SF_{DR} can be completely stated as follows.

Example The assumptions are that (1) the *guideline engine* will recommend the next task for a patient; (2) the transition from one task to another is only possible if the first is connected to the second through the *hasAlternativeTask* property; (3) to move to one of the alternative

Conditions	T (t)	N (n)	M (m)	Task	Recommendation
A	t0 *or* tis	--	--	Action 1	No adjuvant therapy, colonoscopy in 1 year
B	t1 *or* t2	n0	m0	Action 1	No adjuvant therapy, colonoscopy in 1 year
C	t3	--	m0	Action 2	Clinical trial or observation or consider capecitabine or 5-FU/leucovorin
D	t4	n0	m0	Action 3	Capecitabine or 5-FU/leucovorin or FOLFOX or CapeOX
E	t1 *or* t2 *or* t3 *or* t4	n1 *or* n2	m0	Action 4	FOLFOX or CapeOX or FLOX
F	t1 *or* t2 *or* t3 *or* t4	n0 *or* n1 *or* n2	m1	Action 5	Colonoscopy, chest and abdominal/pelvic CT, platelets, chemistry profile

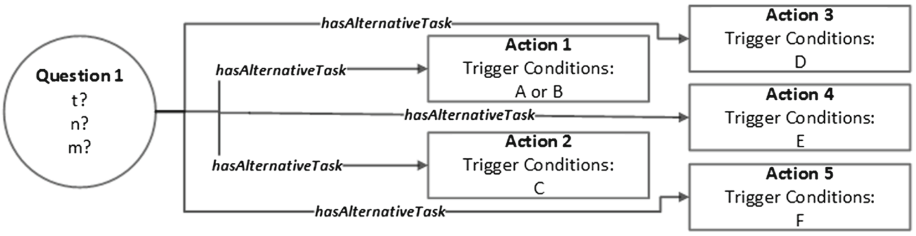


Fig. 2 Clinical setting for the example. The trigger conditions are expressed in terms of three variables, according to the TNM staging system for colon cancer. The possible values for T are $t0, tis, t1, t2, t3$, and $t4$. The possible values for N are $n0$, $n1$, and $n2$. The possible values for M are $m0$ and $m1$

tasks, the trigger conditions of that task must be met; (4) the information necessary to verify the trigger conditions (namely the values for the T, N, and M parameters) will be acquired from an external agent information source, in this case, the oncology information system (*ois*) ; and (6) the *guideline engine* has an SF_{DR} which uses default constraints to continue the execution in the event that there is no answer from *ois*.

The example can be now represented according to the tuple $\langle \Sigma, \mathcal{E}, \Delta, \mathcal{P} \rangle$. In the representation below, a few considerations were made regarding predicate names. The predicate $nt(a, b)$ indicates that b is the task that follows a. The predicate $alt(a, b)$ indicates that b is an alternative task linked to a and reflects the meaning conveyed by the *hasAlternativeTask* property. $tcv(b)$ denotes that the trigger conditions for task b are validated. Predicate names t, n, and m represent their homonymous elements in the TNM classification. In the representation below, every capital letter symbolizes a variable. If the variable is not bounded by a constraint, it is a general purpose variable. For instance, the use of X in the first rule of $\mathcal{P}$ is a way to denote a generic task. The complete representation is as follows:

- $\Sigma = \{ois\}$
- $\mathcal{E} = \{t, n, m\}$
- $\mathcal{P}$ is the following set of rules:
  $nt(X, F) \hookleftarrow \| \; alt(X, F), tcv(F).$
  $tcv(F) \hookleftarrow F \in \{action1\}, T \in \{tis, t0\} \; \| \; t(T)@ois.$
  $tcv(F) \hookleftarrow F \in \{action1\}, T \in \{t1, t2\}, N \in \{n0\}, M \in \{m0\} \; \| \; t(T)@ois, n(N)@ois, m(M)@ois.$
  $tcv(F) \hookleftarrow F \in \{action2\}, T \in \{t3\}, M \in \{m0\} \; \| \; t(T)@ois, m(M)@ois.$

$tcv(F) \hookleftarrow F \in \{action3\}, T \in \{t4\}, N \in \{n0\}, M \in \{m0\} \parallel t(T)@ois, n(N)@ois, m(M)@ois.$
$tcv(F) \hookleftarrow F \in \{action4\}, T \in \{t1, t2, t3, t4\}, N \in \{n1, n2\}, M \in \{m0\} \parallel t(T)@ois, n(N)@ois, m(M)@ois.$
$tcv(F) \hookleftarrow F \in \{action5\}, T \in \{t1, t2, t3, t4\}, N \in \{n0, n1, n2\}, M \in \{m1\} \parallel t(T)@ois, n(N)@ois, m(M)@ois.$
$alt(question1, F) \hookleftarrow F \in \{action1\} \parallel.$
$alt(question1, F) \hookleftarrow F \in \{action2\} \parallel.$
$alt(question1, F) \hookleftarrow F \in \{action3\} \parallel.$
$alt(question1, F) \hookleftarrow F \in \{action4\} \parallel.$
$alt(question1, F) \hookleftarrow F \in \{action5\} \parallel.$

Since, for the current example, the *ois* is the only agent information source, it is the only element in Σ. The *guideline engine*, as the speculative agent, sends questions to the *ois* to know the values for the variables of askable atoms t, n, and m, which are the elements of $\mathcal{E}$. The first rule in $\mathcal{P}$ reflects the primary condition in the example; for a task to be recommended as the next task, it has to be connected to the previous one as an alternative and its trigger conditions have to be validated. The rules that follow represent the trigger conditions for task selection, and the last five rules represent the alternative relationship between *question1* and the other action tasks. However, there is an element missing in the framework, that is the default answer set Δ, which is generated through the procedures for the Generation of default constraints, described in Sect. 4. The execution of the logic program $\mathcal{P}$ starts with a query. In the example, the *guideline engine* would try to answer the query $nt(question1, F)$, in order to determine which task should follow *question1*. An execution trace for this example is provided in Sect. 5.2.

3.2 Preliminary definitions

In order to understand the execution of a CLP $\mathcal{P}$ in the framework, it is important to introduce the notions of goal, reply set, reduction, and derivation. The definitions for these concepts are as follows.

Definition 1 A goal has the form of "$\hookleftarrow C \parallel B_1, \ldots, B_n$" where:

- C is a set of constraints;
- each of $B_1, \ldots, B_n$ is either an atom or an askable atom.

The initial query becomes the first goal in the execution. Following the example described in the previous section, it would be "$\hookleftarrow\parallel nt(question1, F)$". At this point there would be no constraints C for the goal.

Definition 2 A reply set $\mathcal{R}$ for $\mathcal{E}$ is a set of rules of the form "$Q@S \hookleftarrow C \parallel.$", where:

- $Q@S$ is an askable atom;
- each argument of Q is a variable;
- C is a set of constraints over those variables.

Definition 3 A reduction of a goal "$\hookleftarrow C \parallel B_1, \ldots, B_n$" w.r.t. a constraint logic program $\mathcal{P}$, a reply set $\mathcal{R}$ and a subgoal B_i is a goal "$\hookleftarrow C' \parallel B'$" such that:

- there is a rule R in $\mathcal{P} \cup \mathcal{R}$ so that $C \wedge \{B_i = head(R)\} \wedge const(R)$ is consistent and $\{B_i = head(R)\}$ is a conjunction of constraints equal to the arguments of B_i and $head(R)$;

- $C' = C \wedge \{B_i = head(R)\} \wedge const(R)$;
- $B' = B_1, \ldots, B_{i-1}, B_{i+1}, \ldots, B_n \cup body(R)$.

Definition 4 A derivation of a goal "$\hookleftarrow C \parallel B_1, \ldots, B_n$" w.r.t. to a speculative computation constraint framework with default revision $\langle \Sigma, \mathcal{E}, \Delta, \mathcal{P} \rangle$ and a reply set $\mathcal{R}$ is a chain of reductions "$\hookleftarrow C \parallel B_1, \ldots, B_n, \ldots, \hookleftarrow C' \parallel \emptyset$" w.r.t. $\mathcal{P}$ and $\mathcal{R}$, where $\emptyset$ denotes an empty goal. C' is called an answer constraint w.r.t. the goal, the framework and the reply set.

Regarding the execution of a program, a non-askable atom in a goal is reduced into subgoals according to the rules in $\mathcal{P}$. For an askable atom $Q@S$ in a goal, the speculative agent sends a question asking its real value to an agent S in Σ and waits for the answer. Meanwhile, the goal is reduced according to the current reply set $\mathcal{R}$. This set holds the constraints for askable atoms at a given point in the execution. It may contain default constraints, or fact constraints if the speculative agent has already received a reply from S. If no reply is returned, the default constraint is used as a tentative answer. An answer for $Q@S$ is returned in the form of constraints for the variables in the query. A goal is continuously reduced until it becomes empty.

A central notion in the SF_{DR} and its operational model is that of process. It is used to express an alternative computation and a possible path for the resolution of a problem. This concept and others related to it are defined as follows.

Definition 5 An active process is a tuple $\langle \hookleftarrow C \parallel GS, UD \rangle$ in which:

- "$\hookleftarrow C \parallel$" is a set of constraints;
- GS is a set of literals to be proved, called a goal set, and expresses the current status of an alternative computation;
- UD is a set of askable atoms called used defaults, and represents the assumed information about the outside world, i.e., it contains the askable atoms for which default constraints were used in order to reduce a goal in the process.

Definition 6 A suspended process is a tuple $\langle SAS, \hookleftarrow C \parallel GS, UD \rangle$ in which:

- SAS is called a suspended atom set and contains askable atoms $Q@S$. The askable atoms in this set are those whose constraints are responsible for suspending the process;
- GS is a set of literals to be proved, called a goal set, and expresses the current status of an alternative computation;
- UD is a set of askable atoms called used defaults, and represents the assumed information about the outside world, i.e., it contains the askable atoms for which default constraints were used in order to reduce a goal in the process.

Definition 7 A current belief state CBS is a set of rules of the form "$Q@S \hookleftarrow C \parallel$". It contains the beliefs of the speculative agent about the values of askable atoms.

Definition 8 Let $\langle \hookleftarrow C \parallel GS, UD \rangle$ be a process and CBS be a current belief state. A process is active w.r.t. CBS if $C \subseteq CBS$. A process is suspended w.r.t. to CBS otherwise.

Definition 9 APS is the set of active processes.

Definition 10 SPS is the set of suspended processes.

Definition 11 AAQ is the set of already asked questions for askable atoms. AAQ is used to avoid asking redundant questions to agent information sources.

Definition 12 The set of returned facts *RF* contains rules of the form: "$Q@S \hookleftarrow C \,\|$" where $Q@S$ is an askable atom and C is a set of constraints. These rules stand for replies from the agent information sources.

Definition 13 A scenario is an active process $\langle \hookleftarrow C \,\|\, GS, UD\rangle$ in which $UD \neq \emptyset$.

Processes represent alternative ways of computation created when choice points are reached, either by case splitting or default handling. An active process succeeds if, when all the computation is done, there is no reply from the information sources that contradicts the constraints used to reduce askable atoms. A process fails whenever a newly arrived fact constraint contradicts a used default constraint. The mechanisms for the creation, alteration and removal of processes are included in the three phases of Speculative Computation with Default Revision: the *process reduction* phase, the *fact arrival* phase, and the *default revision* phase. The last is a recent addition and the object of study in this work. *Process reduction* is the normal execution of a program, the computation begins with the default rules in Δ and whenever a choice point in the computation is reached, a new process is generated. *Fact arrival phase* occurs when a reply from an agent arrives, it is an interruption phase. After a round of *process reduction* there is the *default revision* phase, in which changes to default constraint rules are assessed and the processes are revised accordingly.

3.3 Process reduction phase

The possible steps of the *process reduction* phase are represented in Figs. 3 and 4. The former corresponds to the very first step of the execution of the framework, while the latter describes the procedures for an iteration step.

The initial step of Fig. 3 coincides with the deployment of the framework, in which a query is submitted and becomes the initial goal set GS. There is only one active process, constructed based on the initial goal set. Since no outside information is known, the current belief state assumes the constraints in the default set. At this point, there are no suspended processes, already asked questions, or returned facts, and, as such, the sets corresponding to these elements are initiated as empty sets. Referring back to the example presented in Sect. 3.1 and the query $nt(question1, F)$, by the iteration step, the first active process would become $\langle\{\hookleftarrow\| \; nt(question1, F)\}, \emptyset\rangle$, with $\{\hookleftarrow\| \; nt(question1, F)\}$ as the initial goal set. The complete procedures for the initial step can be consulted in Algorithm 1 of Annex 2.

The iteration step of Fig. 4 reflects the speculative nature of the framework. Case 1 corresponds to a point in the execution where process reduction, through the derivation of goals (as specified in Definition 4), produced an active process with an empty goal set. When such a process is obtained, its constraints C and used defaults *UD* should be returned. As specified in Definition 14, if *UD* is not an empty set, the process is a scenario.

The procedures under Case 2 correspond to the reduction of a goal (as specified in Definition 3) in the goal set of an active process. There are two cases here. If the goal is a non-askable atom (Case 2.1), then the goal is unified with the head of a rule, a new active process is created replacing the previous in the set of active processes, the constraints in the rule are added to the constraints of the process, and the atoms in the body of the rule are

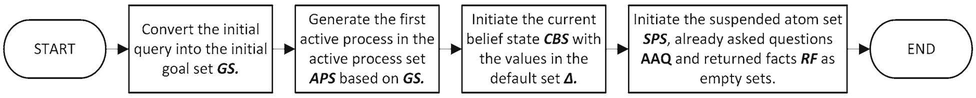

START
Convert the initial query into the initial goal set GS.
Generate the first active process in the active process set APS based on GS.
Initiate the current belief state CBS with the values in the default set Δ.
Initiate the suspended atom set SPS, already asked questions AAQ and returned facts RF as empty sets.
END

Fig. 3 Flowchart with the procedures for the initial step of the *process reduction* phase

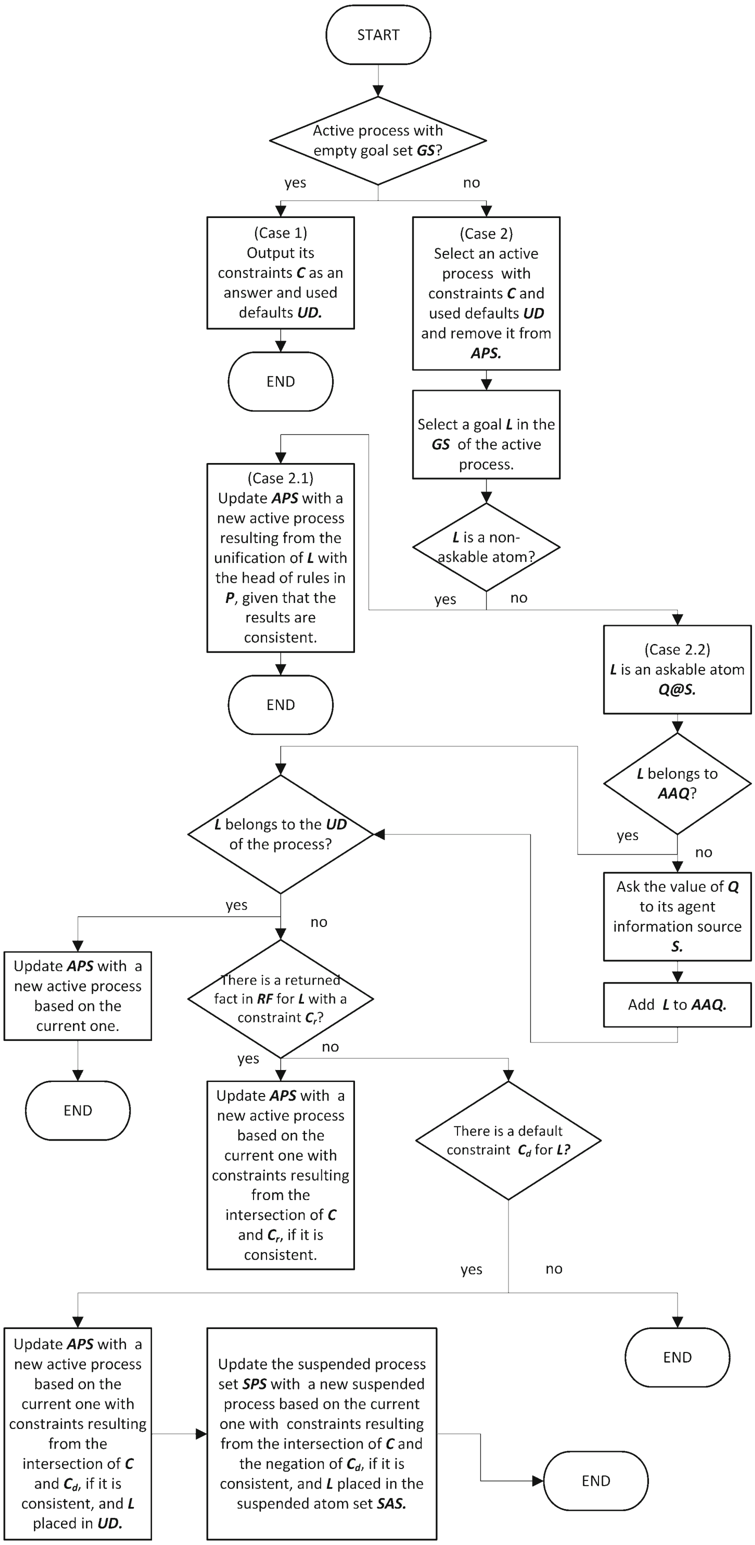

START
Active process with empty goal set ***GS***?
yes
no
(Case 1) Output its constraints ***C*** as an answer and used defaults ***UD.***
END
(Case 2) Select an active process with constraints ***C*** and used defaults ***UD*** and remove it from ***APS.***
Select a goal ***L*** in the ***GS*** of the active process.
L is a non-askable atom?
yes
no
(Case 2.1) Update ***APS*** with a new active process resulting from the unification of ***L*** with the head of rules in ***P***, given that the results are consistent.
END
(Case 2.2) ***L*** is an askable atom ***Q@S.***
L belongs to ***AAQ***?
yes
no
Ask the value of ***Q*** to its agent information source ***S.***
Add ***L*** to ***AAQ.***
L belongs to the ***UD*** of the process?
yes
no
Update ***APS*** with a new active process based on the current one.
END
There is a returned fact in ***RF*** for ***L*** with a constraint C_r?
yes
no
Update ***APS*** with a new active process based on the current one with constraints resulting from the intersection of ***C*** and C_r, if it is consistent.
There is a default constraint C_d for ***L?***
yes
no
Update ***APS*** with a new active process based on the current one with constraints resulting from the intersection of ***C*** and C_d, if it is consistent, and ***L*** placed in ***UD.***
Update the suspended process set ***SPS*** with a new suspended process based on the current one with constraints resulting from the intersection of ***C*** and the negation of C_d, if it is consistent, and ***L*** placed in the suspended atom set ***SAS.***
END
END

Fig. 4 Flowchart with the procedures for the iteration step of the *process reduction* phase

added to the goal set of the process. However, if the goal is an askable atom (Case 2.2), it is necessary to check whether it is in the already asked questions, and thus has already been asked to the agent information sources, and whether it is already a used default, i.e., if it is in the used default set of the process. If this last condition is not fulfilled, the goal is reduced either with a fact constraint in the set of returned facts or with a default constraint. If there is a default constraint available, the constraint is assumed. Upon reduction using default constraints, there are usually two resulting processes, an active process using the default constraint and a suspended process as an alternative that does not use the default. The alternative process is suspended since it is considered to have a lower probability of success.

The complete operational semantics of the iteration step in *process reduction* is disclosed in Algorithm 2 of Annex 2 with an exact correspondence of case numbers to those of Fig. 4.

3.4 Fact arrival phase

Suppose a constraint is returned from an agent denoting an information source S for a question $Q@S$. This triggers a *fact arrival* phase, whose procedures are outlined in Fig. 5. The returned constraint is denoted as "$Q@S \hookleftarrow C_r \parallel$".

In *fact arrival*, the set of returned facts *RF*, which keeps all the replies from the agent information sources, is updated with the fact constraint. The *CBS* is also updated with the fact returned from *S*, which means the old default constraint is replaced with the fact constraint. After that, three cases are considered.

In Case 1 of Fig. 5, the reply entails the default constraint, and, as such, the processes using the default in the set of active processes *APS* are replaced with updated versions. These updated versions take into account the new fact constraint and carry out the execution trace in their origin. The set of suspended processes *SPS* is revised as well. Suspended processes which use the default constraint are also replaced with updated versions. Additionally, the processes suspended because of the default constraint, identified as the ones having the corresponding predicate in the suspended atom set *SAS*, are removed from the computation.

In Case 2, the reply contradicts the default constraint. The active processes using the default are removed and the suspended processes which are consistent with the new fact constraint are resumed. Having a *default revision* phase has repercussions on *fact arrival*. As a product of *default revision*, there can be suspended processes with multiple suspended askable atoms, and thus it is necessary to check whether, after *fact arrival*, the *SAS* of a suspended process becomes an empty set or not. If it does, the process can be resumed, otherwise it remains suspended.

Finally, in Case 3, the reply does not entail or contradict the default constraint, but is consistent with it. Only the active processes which are consistent with the new fact constraint are kept as updated versions in *APS*. Similarly, only the suspended processes which are consistent with the new fact constraint are activated and added as updated versions to *APS*. All the others are removed from the computation.

The complete operations for fact arrival are described in Algorithm 3 in Annex 2.

3.5 Default revision phase

When a fact arrives, the Generation of defaults (presented in Sect. 4) produces a set containing only the changed default constraint rules. This set is referred to as *New*Δ and each of its members is a new default denoted as "$Q_d@S \hookleftarrow C_{newd} \parallel$.". *Default revision* takes place after *fact arrival* and a round of *process reduction*, based on *New*Δ. This phase denotes a

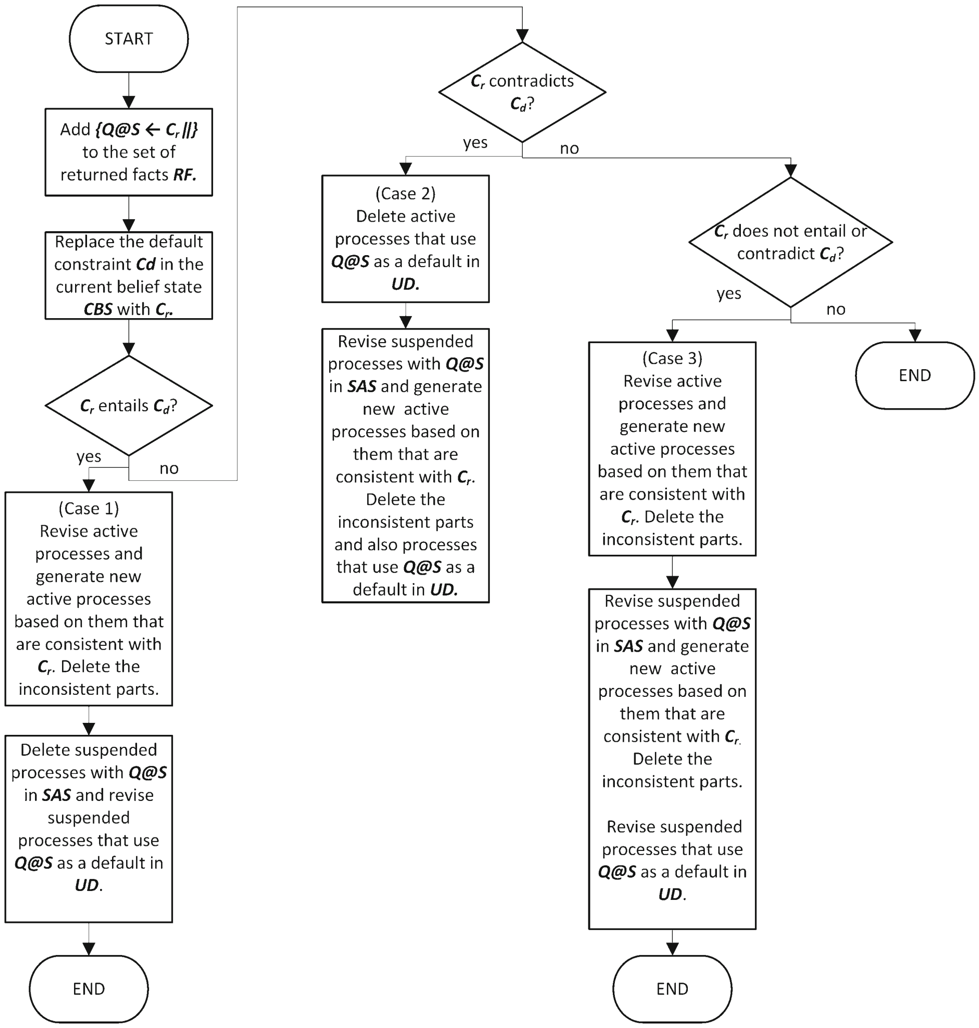

START
Add {$Q@S \leftarrow C_r \|$} to the set of returned facts *RF.*
Replace the default constraint *Cd* in the current belief state *CBS* with C_r.
C_r entails C_d?
yes
no
(Case 1) Revise active processes and generate new active processes based on them that are consistent with C_r. Delete the inconsistent parts.
Delete suspended processes with *Q@S* in *SAS* and revise suspended processes that use *Q@S* as a default in *UD*.
END
C_r contradicts C_d?
yes
no
(Case 2) Delete active processes that use *Q@S* as a default in *UD.*
Revise suspended processes with *Q@S* in *SAS* and generate new active processes based on them that are consistent with C_r. Delete the inconsistent parts and also processes that use *Q@S* as a default in *UD.*
C_r does not entail or contradict C_d?
yes
no
(Case 3) Revise active processes and generate new active processes based on them that are consistent with C_r. Delete the inconsistent parts.
Revise suspended processes with *Q@S* in *SAS* and generate new active processes based on them that are consistent with C_r. Delete the inconsistent parts.
Revise suspended processes that use *Q@S* as a default in *UD*.
END
END

Fig. 5 Flowchart with the procedures for the *fact arrival* phase

change in the other default constraints as a result of *fact arrival*. For this to occur, *New*Δ cannot be an empty set.

The default revision phase consists in changing all the processes according to the new default constraints provided by the method for the generation of defaults. Active and suspended processes are revised in order to determine if their constraints are consistent with the new default constraint. According to the position of the atom attached to the new default, i.e., if it is a suspended atom or a used default, the existing processes may generate new active and suspended processes, but their execution trace is never removed, unlike what happens in the *fact arrival* phase. This keeps the scenarios coherent with the newly arrived fact constraints. It is an on-line dynamic mechanism because it responds to the replies from the outside agent information sources, updates the default constraints used in the computation, and keeps the processes consistent with those updates.

The procedures for *default revision* are shown in Fig. 6. Its first operation is to update the *CBS* so that future *process reduction* phases take the changed defaults into account.

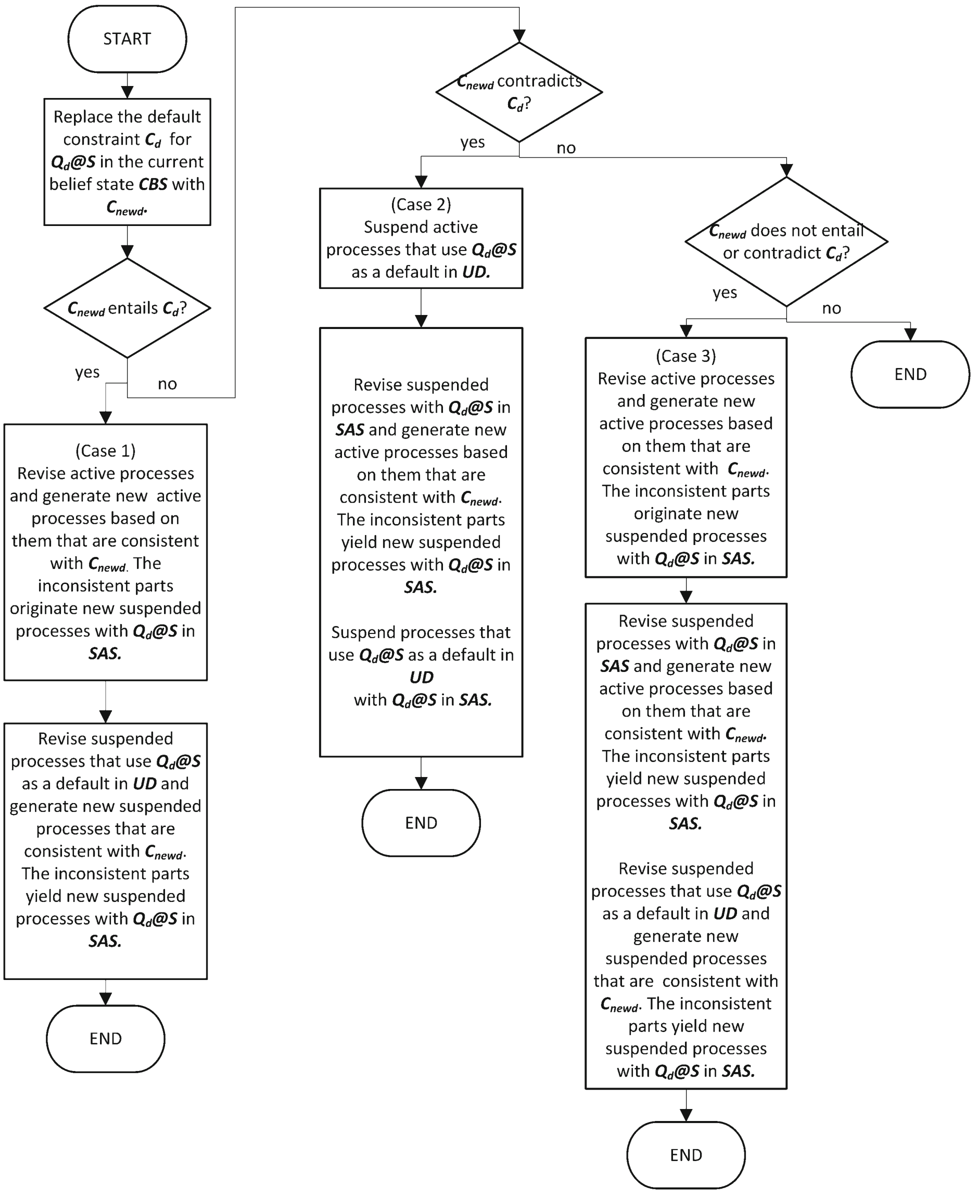

START
Replace the default constraint C_d for Q_d@S in the current belief state CBS with C_{newd}.
C_{newd} entails C_d?
yes
no
(Case 1) Revise active processes and generate new active processes based on them that are consistent with C_{newd}. The inconsistent parts originate new suspended processes with Q_d@S in SAS.
Revise suspended processes that use Q_d@S as a default in UD and generate new suspended processes that are consistent with C_{newd}. The inconsistent parts yield new suspended processes with Q_d@S in SAS.
END
C_{newd} contradicts C_d?
yes
no
(Case 2) Suspend active processes that use Q_d@S as a default in UD.
Revise suspended processes with Q_d@S in SAS and generate new active processes based on them that are consistent with C_{newd}. The inconsistent parts yield new suspended processes with Q_d@S in SAS.
Suspend processes that use Q_d@S as a default in UD with Q_d@S in SAS.
END
C_{newd} does not entail or contradict C_d?
yes
no
(Case 3) Revise active processes and generate new active processes based on them that are consistent with C_{newd}. The inconsistent parts originate new suspended processes with Q_d@S in SAS.
Revise suspended processes with Q_d@S in SAS and generate new active processes based on them that are consistent with C_{newd}. The inconsistent parts yield new suspended processes with Q_d@S in SAS.
Revise suspended processes that use Q_d@S as a default in UD and generate new suspended processes that are consistent with C_{newd}. The inconsistent parts yield new suspended processes with Q_d@S in SAS.
END
END

Fig. 6 Flowchart with the procedures for the *default revision* phase

In Case 1 of Fig. 6, the new default constraint entails the old default constraint, and, as such, the processes using the old default in the set of active processes *APS* are replaced with updated versions. These updated versions take into account the new default constraint and carry out the execution trace in their origin. The set of suspended processes *SPS* is revised as well. Suspended processes which use the old default constraint are also replaced with updated versions. However, it is important to note that the portions of active and suspended processes that are consistent with the negation of the new default constraint (the same is to say inconsistent with the new default constraint) are still kept in the computation as suspended processes.

In Case 2 of Fig. 6, the new default constraint contradicts the old default constraint. The active processes using the old default become suspended and the suspended processes which are consistent with the new default constraint are resumed. However, suspended processes or parts of suspended processes which are consistent with the negation of the new default constraint are still kept in the computation as updated suspended versions.

Finally, in Case 3 of Fig. 6, the new default constraint does not entail or contradict the old, but is consistent with it. In this case both active and suspended processes are revised, and the parts of these processes which are consistent with the new default constraint continue or become active in the form of updated versions, while the parts that are consistent with the negation of the new default constraint are suspended.

A curious aspect of *default revision*, and a major difference from previous works [9,21,22,24], is that processes may be suspended due to default constraints about more than one askable atom. This happens when there is a revision of the *SPS* for a new default constraint that contradicts the old one. A process may be in a suspended state due to constraints about a variable unrelated to the new default, and the old default constraint may already be part of the process, which means that the corresponding askable atom is in the set of used defaults *UD*. In such a situation, it is necessary to remove that askable atom and add it to the suspended atom set, which means that the process is now suspended due to an additional constraint. As a result, it became necessary to perform additional operations for the management of askable atoms in *fact arrival*, since a process can only be resumed if it has an empty suspended atom set.

The complete operational semantics for the *default revision* phase is shown in the Algorithms 4 and 5 in Annex 2. There is an exact correspondence between the cases in Fig. 6 and the cases featured in the algorithms.

3.6 Correctness of the operational model

The following claim shows the correctness of the operational model presented in Sects. 3.3, 3.4, and 3.5:

Claim 1 *Let SF_{DR} be a Speculative Computation Framework with Default Revision. Let P be an ordinary process, GS_{init} be an initial goal set, UD a set of askable atoms used as defaults, C an answer constraint obtained from the operational model, and RF a set of rules returned from the other information sources when a reply arrives. Then, there exists an answer constraint C' w.r.t. the constraint framework $\langle \Sigma, \mathcal{P} \rangle$ and the reply set $RF \cup \{\delta(Q@S) | Q@S \in UD\}$ s.t. $\pi_V(C)$ entails $\pi_V(C')$ where V is the set of variables that occur in GS_{init}, and π_V is the projection of constraints onto V.*

A sketch of proof for Claim 1 can be consulted in Annex 3.

4 Generation of default constraints

The objective of this component of Speculative Computation with Default Revision is to produce the set of default constraint rules Δ and the set of changed default constraint rules $New\Delta$.

The generation of default constraints encompasses two different types of procedures, both described in Fig. 7. The first is the Learning of the Default Model and it consists of steps to produce a BN model from which it is later possible to extract the default constraints. This extraction happens in the second procedure, the Collection of Defaults, which has two

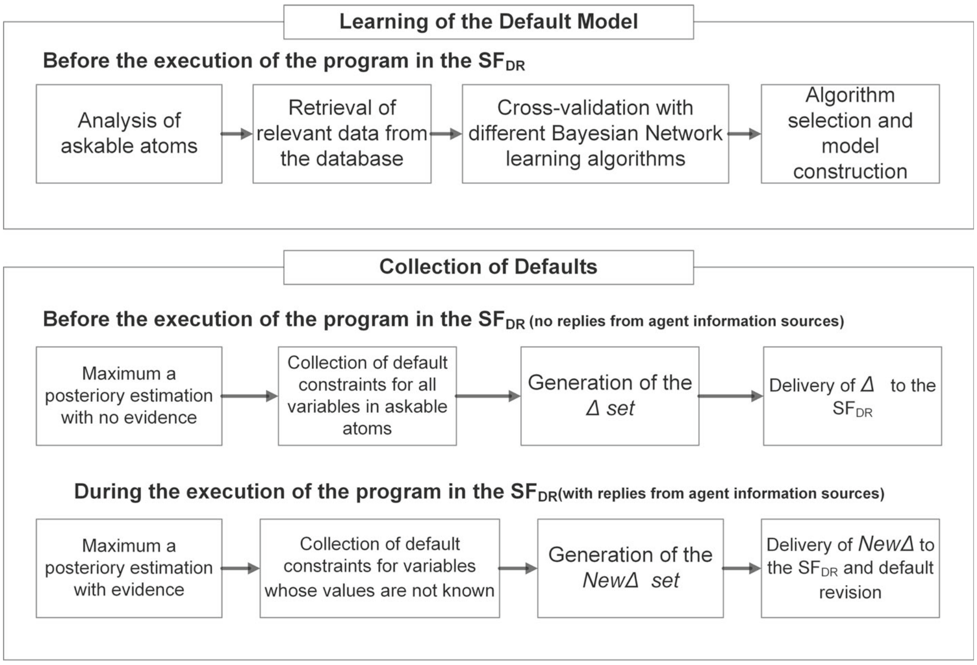

Learning of the Default Model
Before the execution of the program in the SF_DR
Analysis of askable atoms
Retrieval of relevant data from the database
Cross-validation with different Bayesian Network learning algorithms
Algorithm selection and model construction
Collection of Defaults
Before the execution of the program in the SF_DR (no replies from agent information sources)
Maximum a posteriory estimation with no evidence
Collection of default constraints for all variables in askable atoms
Generation of the Δ set
Delivery of Δ to the SF_DR
During the execution of the program in the SF_DR(with replies from agent information sources)
Maximum a posteriory estimation with evidence
Collection of default constraints for variables whose values are not known
Generation of the NewΔ set
Delivery of NewΔ to the SF_DR and default revision

Fig. 7 Procedures in the Generation of Default Constraints. The two types of procedures, Learning of the Default Model and Collection of Defaults, comprise steps to construct a probabilistic model with the variables in the problem the speculative agent has to address and to collect the default constraints from that model

different sets of steps, depending on the moment of speculative computation the execution is at. Although a part of Speculative Computation with Default Revision, the Generation of Default Constraints is exterior to the operational model of the SF_{DR} described in Sect. 4 and assumes a supportive role.

The following sections describe our reasoning for choosing BNs as the underlying default model and the different steps in the Generation of Defaults.

4.1 Choosing the underlying default model

In order to extract a meaningful set of default constraints that can fill in information gaps, the default generation method has to be data-driven. Additional requirements of the method are that it should be capable of finding existing dependence relationships between the variables of askable atoms in $\mathcal{P}$ so as to produce the most likely set of default constraints, and be transparent in the sense that it conveys those relationships in a clear way to both humans and machines. Within the scope of predictive modelling, the problem at hand deviates from a classification or regression problem because there is no target class/variable. Instead, the goal is to find the most probable collective state of information, i.e., the most likely values for variables whose real values are unknown. These requirements excluded predictive models such as Neural Networks or Support Vector Machines [32].

Taking these aspects into account, the attention turned to graphical probabilistic models, and more specifically to BNs, which are able to represent this information. With BNs it is possible to update the calculations according to evidences, which fits the kind of dynamic revision mechanism required for Speculative Computation with Default Revision. Furthermore, it is possible to establish relationships of cause and effect as opposed to undirected models such as pure Markov Networks. The works in [30] and [29] reflect the main use of BNs in medicine, namely in the identification of disease dynamics for diagnosis, prognosis or sudden changes in the state of a patient. Although the medical field is a prominent domain of application for BNs, they have also been used in many other domains to model dependability and perform risk analysis and maintenance. For a comprehensive review please consult [31]. However, in the work proposed here, BNs are used as a support for speculative computation in order to generate default constraints, which is a new application of this knowledge representation model.

4.2 Learning of the default model

The Learning of the Default Model in Fig. 7 comprises four sequential steps. The first is the analysis of askable atoms, in which the speculative agent analyses the logic program component $\mathcal{P}$ of the SF_{DR} and identifies the relevant variables for speculative computation. In the example provided in Sect. 3.1, the *guideline engine* would identify t, n, and m as askable atoms and T, N, and M as relevant variables.

These variables are then used in the second step, the retrieval of relevant data from the database. The speculative agent retrieves data of previous cases about the identified variables from a local information repository. Referring again to the example in Sect. 3.1, the *guideline engine* would use the data about previous executions of the clinical guideline in the *local repository*.

The following step is to perform cross-validation with different BN learning algorithms on the retrieved data set in order to determine which one will generate the model with the best predictions on future data. BN learning algorithms can be divided into two groups according to their search strategy [25]: score-based learning and constraint-based learning. The former assign a score to each candidate BN and try to maximize it with an heuristic search, while the latter learn the network structure by analysing the probabilistic relations entailed by the Markov property of BNs with conditional independence tests. As this is not the main topic of the paper, additional details about structure learning are provided in [25]. The set of learning algorithms used in this step includes: two score-based learning algorithms, the Hill–Climbing (*hc*) and the Tabu-Search (*tabu*); three constraint-based search algorithms, the Grow–Shrink (*gs*), the Incremental Association (*iamb*), and the Chow-Liu (*chowliu*); and one hybrid algorithm, the Max–Min Hill–Climbing (*mmhc*). The loss function used in cross-validation is the *Log Likelihood Loss* (*logl*), typically applied to this kind of problem and defined as in Eq. 1 [8]:

$$logl = -log[P(D|G)] \tag{1}$$

where D is the data used to learn the network structure and G is the graph structure produced by the algorithm. The *logl* provides a measure of the entropy that a model exports in order to keep its own entropy low. The target is to minimize this value and choose the model with the lowest *logl* [8]. The *logl* is a common way to compare how well a distribution for a model fits the data. However, using the *logl* as a scoring function in learning BNs has the disadvantage of producing networks that are too complex and slightly overfitted. But the *logl* is at the core of other functions, such as the Akaike Information Criterion, the Bayesian Information Criterion, and so forth, which have penalties to compensate for that and produce more sparse structures [13]. The use of *logl* herein is intended as a demonstration of the role played by the scoring function in the generation of default constraints and, as such, the simplest measure was used.

With the results from cross-validation, the best algorithm is selected and with it the BN is constructed in the last step of the Learning of the Default Model. The whole procedure takes place at a point in time before the execution of a program in the SF_{DR} and outputs a BN ready for the Collection of Defaults. Going back to the example of Sect. 3.1, the output would be a BN that establishes a relationship between the T, N, and M variables.

4.3 Collection of defaults

The Collection of Defaults is a procedure by which the BN produced by the Learning of the Default Model is conditioned in order to obtain default constraints. It is based on the *maximum a posteriori* estimation (MAP), a form of posterior distribution in Bayesian statistics that allows one to obtain a point estimation of unobserved variables based on collected evidence. It is defined as in Eq. 2 [11]:

$$\theta_{MAP} = \max_{\theta} \; P(\theta|e) \tag{2}$$

where θ represents the goal variables for which the estimation is calculated and e represents the available evidence. A MAP calculates the values of the unobserved variables that maximize the probability distribution, providing the most likely setting for the variables in the network.

The first set of steps of the Collection of Defaults described in Fig. 7 occurs before the execution of a program in the SF_{DR} by the speculative agent. It corresponds to the initial MAP estimation, in which there is no reply from the agent information sources, and thus no evidence to submit. In the MAP estimation the goal variables correspond to all the variables in askable atoms. The result is a set of values for each variable that, once retrieved, are put into the form of constraint rules like "$Q@S \hookleftarrow C \; \|$." where Q is an askable atom, S is an agent information source and C is a constraint. The constraints are used to build the default answer set Δ which is then delivered to the SF_{DR}.

The second set of steps in Fig. 7 occurs during the execution of a program in the SF_{DR}. The speculative agent keeps track of all the replies from the agent information sources and, when a fact constraint arrives with true information about a variable in an askable atom, it performs a MAP estimation using the new fact and all the previous ones as evidence in order to obtain an estimation for the yet unobserved variables. If there are no previous replies, the newly arrived fact is the only evidence in the MAP. The resulting values are then converted to constraints and compared with the existing default constraints for the respective askable atoms. Those that are different are put into a new default answer set $New\Delta$ and then delivered to the SF_{DR}, triggering a *default revision* phase. This procedure takes place every time a fact arrives.

The Collection of Defaults and the *default revision* phase provide agents with a form of on-line revision capability. They ensure that the scenarios represented by active processes get progressively closer to the actual state of the real world. This is an advantage for agents which have to deal with problems that demand an outcome, even in the presence of incomplete information.

5 Example: clinical decision support

The principles of Speculative Computation with Default Revision can be applied to various domains and tasks. To demonstrate the formalization of a specific situation, an example in clinical decision support was provided in Sect. 3.1. Now, the same example will be used to demonstrate how the default set Δ can be obtained in order to complete the SF_{DR}, based on the steps outlined in Sect. 4. Afterwards there is a step by step execution of the example in the framework. Incomplete information in the example may be due to the difficulty in determining the true value for a variable or problems in the communication between the *guideline engine*, which assumes the role of the speculative agent, and *ois*, with the role of agent information source. Another possibility is that the *ois* does not possess the information required by the *guideline engine*.

5.1 Default model and initial set of default constraints

The first procedure to be applied from the Generation of Defaults, before the execution of the program in the SF_{DR} and any attempt from the *guideline engine* to solve the problem in the program, is the Learning of the Default Model. As stated above, the objective is to obtain a BN with the variables that are featured in the problem the speculative agent has to solve. From the analysis of askable atoms t, n, and m in the SF_{DR} of Section 3.1, it is necessary to extract data from the *local repository* about variables T, N, and M. Following the example provided above, data about TNM staging of 515 patients were used for the Learning of the Default Model and to assess the process. These patients had surgery to the colon and underwent colon cancer treatment following the above-mentioned guideline [1] at the Hospital of Braga, in Portugal.

In order to estimate the performance of the predictive model, 5-fold cross-validation was used, and its results, in terms of *logl* are shown in Fig. 8. The algorithms with the worst performance were the score-based, while the constraint-based and the hybrid algorithms performed fairly better. Given that the *iamb* is the one with the lowest *logl* ($logl = 1.815$), it was selected to construct the BN, yielding the network represented in Fig. 9. As can be seen, the algorithm was able to establish a dependence relationship between T and N, but not with M, which means that M remains unaffected by changes in the other two variables. This is particularly useful for it indicates to the speculative agent that tumor invasion and local lymph node metastases might be correlated, but distant metastases are not dependent on the other two. This type of conclusion is something the BN is capable of providing and has an impact in the generation of default constraints. For instance, it is possible to know that

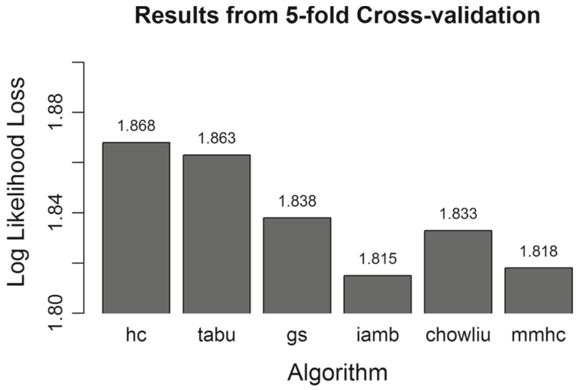


Fig. 8 Results from 5-fold cross-validation for the six BN learning algorithms in terms of log likelihood loss. The set of learning algorithms used in this step includes: two score-based learning algorithms, the Hill–Climbing (*hc*) and the Tabu-Search (*tabu*); three constraint-based search algorithms, the Grow–Shrink (*gs*), the Incremental Association (*iamb*), and the Chow–Liu (*chowliu*); and one hybrid algorithm, the Max–Min Hill–Climbing (*mmhc*)

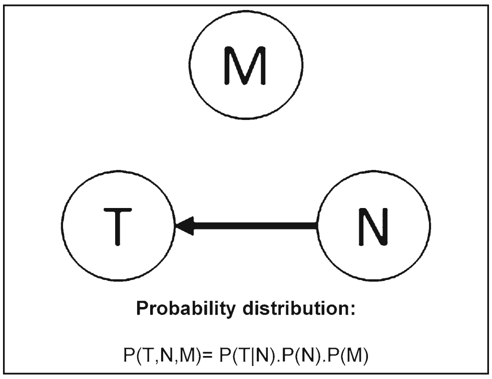


Fig. 9 Network structure produced by the *iamb* learning algorithm about the TNM staging of colon cancer and respective probability distribution

if a value for T is known to be true, it might affect the default constraint of N, but it will not affect the default constraint of M.

With the BN of Fig. 9, it is possible to proceed to the Collection of Defaults in order to generate the initial set of default constraints, before the execution of the program in the SF_{DR}. The goal variables are T, N, and M. After performing the MAP estimation with no evidence on the network, the values obtained were respectively $T = t3$, $N = n2$, and $M = m1$, with $TNM_{MAP} = \max_{T,N,M} \; P(T, N, M \mid \emptyset) \approx 0.4395$. These will be the values used as initial default constraints in speculative computation. The default answer set is thus defined as follows:

- Δ is the following set of rules:
  $t(T)@ois \hookleftarrow T \in \{t3\} \parallel .$
  $n(N)@ois \hookleftarrow N \in \{n2\} \parallel .$
  $m(M)@ois \hookleftarrow M \in \{m1\} \parallel .$

The Δ set is added to the SF_{DR} defined for the example in Sect. 3.1 and now all the elements of the framework are gathered. The network is later used during the execution phases in order to derive new default constraints according to newly arrived facts.

All the resources for the learning and evaluation of BNs mentioned above are available in the *bnlearn* library for R [25], and the resources for the MAP estimation are available in the Java *inflib* library from the SamIam project [4].

5.2 Execution of the example and discussion

Now, based on the formalization provided in Sect. 3.1 and predictive model provided in Sect. 5.1, the execution of the program in the example is shown. The starting point is the already specified $SF_{DR} = \langle \Sigma, \mathcal{E}, \Delta, \mathcal{P} \rangle$. The operations in Sects. 3.3, 3.4, and 3.5 are applied to the program in $\mathcal{P}$.

A goal in the goal set GS at a leftmost position from a newly created or newly resumed process is always selected for reduction. In the representation, the selected goal is underlined, and, if a set remains unchanged from one step to another, it is omitted. The following steps represent the execution trace for the query $nt(question1, F)$, which aims to determine which task should be recommended after *question1* in the management of a patient with colon cancer:

1. **Process Reduction, Initial Step (Fig. 3):**
   - The query originates the first active process and takes the place of its first goal for reduction;

- There are no suspended processes, already asked questions, or returned facts, and, as such, the corresponding sets are empty;
- The current belief state assumes the constraint rules in the default answer set.

$APS = \Big\{\langle\{\hookleftarrow\| \ \underline{nt(question1, F)}\}, \emptyset\rangle\Big\}$
$SPS = \emptyset$
$AAQ = \emptyset$
$RF = \emptyset$
$CBS = \{$
$t(T)@ois \hookleftarrow T \in \{t3\} \|,$
$n(N)@ois \hookleftarrow N \in \{n2\} \|,$
$m(M)@ois \hookleftarrow M \in \{m1\} \|$
$\}$

2. **Process Reduction, by Case 2.1 (Fig. 4):**

   - *nt*(*question*1, *F*) is not an askable atom and, as such, it unifies with the head of a rule in $\mathcal{P}$;
   - The body of the rule replaces *nt*(*question*1, *F*) in the goal set of the updated version of the active process;
   - Since *alt*(*question*1, *F*) is the only element in the goal set, it is selected for the next reduction step.

   $APS = \{\langle\{\hookleftarrow\| \ \underline{alt(question1, F)}, tcv(F)\}, \emptyset\rangle\}$

3. **Process Reduction, by Case 2.1 (Fig. 4):**

   - *alt*(*question*1, *F*) is not an askable atom and, as such, it unifies with the head of five rules in $\mathcal{P}$, splitting the active process into five new active processes;
   - The constraints of the rules are added to the constraints of the processes they originated;
   - Since *tcv*(*F*) is at a leftmost position in the goal set, it selected for the next reduction step in the active processes.

   $APS = \{$
   $\langle\{\hookleftarrow F \in \{action1\} \ \| \ \underline{tcv(F)}\}, \emptyset\rangle,$
   $\langle\{\hookleftarrow F \in \{action2\} \ \| \ \underline{tcv(F)}\}, \emptyset\rangle,$
   $\langle\{\hookleftarrow F \in \{action3\} \ \| \ \underline{tcv(F)}\}, \emptyset\rangle,$
   $\langle\{\hookleftarrow F \in \{action4\} \ \| \ \underline{tcv(F)}\}, \emptyset\rangle,$
   $\langle\{\hookleftarrow F \in \{action5\} \ \| \ \underline{tcv(F)}\}, \emptyset\rangle$
   $\}$

4. **Process Reduction, by Case 2.1 (Fig. 4):**

   - *tcv*(*F*) is not an askable atom and, as such, it unifies with the head of rules in $\mathcal{P}$, splitting the active processes into new active processes according to the consistency of the constraints in the process and the constraints in the rules;
   - The body of the rules replaces *tcv*(*F*) in the goal set of the updated versions of the active processes and the constraints of the rules are added to the processes;
   - Since $t(T)@ois$ is at a leftmost position in the goal set of active processes, it selected for the next reduction step.

   $APS = \{$
   $\langle\{\hookleftarrow F \in \{action1\}, T \in \{tis, t0\} \ \| \ \underline{t(T)@ois}\}, \emptyset\rangle,$

$\langle\{\hookleftarrow F \in \{action1\}, T \in \{t1, t2\}, N \in \{n0\}, M \in \{m0\} \parallel \underline{t(T)@ois}, n(N)@ois, m(M)@ois\}, \emptyset\rangle$,
$\langle\{\hookleftarrow F \in \{action2\}, T \in \{t3\}, M \in \{m0\} \parallel \underline{t(T)@ois}, m(M)@ois\}, \emptyset\rangle$,
$\langle\{\hookleftarrow F \in \{action3\}, T \in \{t4\}, N \in \{n0\}, M \in \{m0\} \parallel \underline{t(T)@ois}, n(N)@ois, m(M)@ois\}, \emptyset\rangle$,
$\langle\{\hookleftarrow F \in \{action4\}, T \in \{t1, t2, t3, t4\}, N \in \{n1, n2\}, M \in \{m0\} \parallel \underline{t(T)@ois}, n(N)@ois, m(M)@ois\}, \emptyset\rangle$,
$\langle\{\hookleftarrow F \in \{action5\}, T \in \{t1, t2, t3, t4\}, N \in \{n0, n1, n2\}, M \in \{m1\} \parallel \underline{t(T)@ois}, n(N)@ois, m(M)@ois\}, \emptyset\rangle$
}

5. **Process Reduction, by Case 2.2 (Fig. 4):**
   $t(T)$ is asked to ois and since $(t(T)@ois \hookleftarrow T \in \{t3\} \parallel) \in \Delta$):

   - $t(T)@ois$ is added to AAQ denoting that a question for this askable atom has already been sent;
   - Since the CBS holds the default constraint rule $(t(T)@ois \hookleftarrow T \in \{t3\} \parallel)$, the active processes are reduced with it;
   - The processes or parts of processes which are consistent with the default constraint remain active as updated versions with $t(T)@ois$ in the used default set;
   - The processes or parts of processes which are inconsistent with the default constraint become suspended as updated versions with $t(T)@ois$ in the suspended atom set;
   - Since $n(N)@ois$ is at a leftmost position in the goal set of active processes, it selected for the next reduction step.

   $APS = \{$
   $\langle\{\hookleftarrow F \in \{action2\}, T \in \{t3\}, M \in \{m0\} \parallel m(M)@ois\}, \{t(T)@ois\}\rangle$,
   $\langle\{\hookleftarrow F \in \{action4\}, T \in \{t3\}, N \in \{n1, n2\}, M \in \{m0\} \parallel \underline{n(N)@ois}, m(M)@ois\}, \{t(T)@ois\}\rangle$,
   $\langle\{\hookleftarrow F \in \{action5\}, T \in \{t3\}, N \in \{n0, n1, n2\}, M \in \{m1\} \parallel \underline{n(N)@ois}, m(M)@ois\}, \{t(T)@ois\}\rangle$
   }
   $SPS = \{$
   $\langle\{\underline{\{t(T)@ois\}}, \hookleftarrow F \in \{action1\}, T \in \{tis, t0\} \parallel \emptyset\}, \emptyset\rangle^{P_1}$,
   $\langle\{\underline{\{t(T)@ois\}}, \hookleftarrow F \in \{action1\}, T \in \{t1, t2\}, N \in \{n0\}, M \in \{m0\} \parallel n(N)@ois, m(M)@ois\}, \emptyset\rangle^{P_2}$,
   $\langle\{\underline{\{t(T)@ois\}}, \hookleftarrow F \in \{action3\}, T \in \{t4\}, N \in \{n0\}, M \in \{m0\} \parallel n(N)@ois, m(M)@ois\}, \emptyset\rangle^{P_3}$,
   $\langle\{\underline{\{t(T)@ois\}}, \hookleftarrow F \in \{action4\}, T \in \{t1, t2, t4\}, N \in \{n1, n2\}, M \in \{m0\} \parallel n(N)@ois, m(M)@ois\}, \emptyset\rangle^{P_4}$,
   $\langle\{\underline{\{t(T)@ois\}}, \hookleftarrow F \in \{action5\}, T \in \{t1, t2, t4\}, N \in \{n0, n1, n2\}, M \in \{m1\} \parallel n(N)@ois, m(M)@ois\}, \emptyset\rangle^{P_5}$
   }
   $AAQ = \{t(T)@ois\}$
6. **Process Reduction, by Case 2.2 (Fig. 4):**
   $n(N)$ is asked to ois and since $(n(N)@ois \hookleftarrow N \in \{n2\} \parallel) \in \Delta$):

   - $n(N)@ois$ is added to AAQ denoting that a question for this askable atom has already been sent;

- Since the *CBS* holds the default constraint rule $(n(N)@ois \hookleftarrow N \in \{n2\} \parallel)$, the active processes are reduced with it;
- The processes or parts of processes which are consistent with the default constraint remain active as updated versions with $n(N)@ois$ in the used default set;
- The processes or parts of processes which are inconsistent with the default constraint become suspended as updated versions with $n(N)@ois$ in the suspended atom set;
- Since $m(M)@ois$ is the only element in the goal set of active processes, it selected for the next reduction step.

$APS = \{$
$\langle\{\hookleftarrow F \in \{action2\}, T \in \{t3\}, M \in \{m0\} \parallel \underline{m(M)@ois}\}, \{t(T)@ois\}\rangle,$
$\langle\{\hookleftarrow F \in \{action4\}, T \in \{t3\}, N \in \{n2\}, M \in \{m0\} \parallel \underline{m(M)@ois}\}, \{t(T)@ois, (N)$
$@ois\}\rangle,$
$\langle\{\hookleftarrow F \in \{action5\}, T \in \{t3\}, N \in \{n2\}, M \in \{m1\} \parallel \underline{m(M)@ois}\}, \{t(T)@ois, n(N)$
$@ois\}\rangle$
$\}$
$SPS = \{$
$\langle\{\underline{\{n(N)@ois\}}, \hookleftarrow F \in \{action4\}, T \in \{t3\}, N \in \{n1\}, M \in \{m0\} \parallel m(M)@ois\}, \{t(T)$
$@ois\}\rangle^{P_6},$
$\langle\{\underline{\{n(N)@ois\}}, \hookleftarrow F \in \{action5\}, T \in \{t3\}, N \in \{n0, n1\}, M \in \{m1\} \parallel$
$m(M)@ois\}, \{t(T)@ois\}\rangle^{P_7},$
P_1, P_2, P_3, P_4, P_5
$\}$
$AAQ = \{t(T)@ois, n(N)@ois\}$

7. **Process Reduction, by Case 2.2 (Fig. 4):**
   $m(M)$ is asked to *ois* and since $((m(M)@ois \hookleftarrow M \in \{m1\} \parallel) \in \Delta)$:

   - $m(M)@ois$ is added to *AAQ* denoting that a question for this askable atom has already been sent;
   - Since the *CBS* holds the default constraint rules $(m(M)@ois \hookleftarrow M \in \{m1\} \parallel)$, the active processes are reduced with it;
   - The processes or parts of processes which are consistent with the default constraint remain active as updated versions with $m(M)@ois$ in the used default set;
   - The processes or parts of processes which are inconsistent with the default constraint become suspended as updated versions with $m(M)@ois$ in the suspended atom set;

   $APS = \{$
   $\langle\{\hookleftarrow F \in \{action5\}, T \in \{t3\}, N \in \{n2\}, M \in \{m1\} \parallel \emptyset\}, \{t(T)@ois,$
   $n(N)@ois, m(M)@ois\}\rangle^{P_8}$
   $\}$
   $SPS = \{$
   $\langle\{\underline{\{m(M)@ois\}}, \hookleftarrow F \in \{action2\}, T \in \{t3\}, M \in \{m0\} \parallel \emptyset\}, \{t(T)@ois\}\rangle^{P_9},$
   $\langle\{\underline{\{m(M)@ois\}}, \hookleftarrow F \in \{action4\}, T \in \{t3\}, N \in \{n2\}, M \in \{m0\} \parallel \emptyset\},$
   $\{t(T)@ois, n(N)@ois\}\rangle^{P_{10}},$
   $P_1, P_2, P_3, P_4, P_5, P_6, P_7$
   $\}$
   $AAQ = \{t(T)@ois, n(N)@ois, m(M)@ois\}$
8. **Process Reduction, by Case 1 (Fig. 4):**
   $APS = \{$

$\langle\{\hookleftarrow F \in \{action5\}, T \in \{t3\}, N \in \{n2\}, M \in \{m1\} \parallel \emptyset\}, \{t(T)@ois, n(N)@ois, m(M)@ois\}\rangle^{P_8}$
}
$C = \{F \in \{action5\}, T \in \{t3\}, N \in \{n2\}, M \in \{m1\}\}$
$UD = \{t(T)@ois, n(N)@ois, m(M)@ois\}$

Up until step 4 *process reduction* occurs with the unification of the goals in the GS with the head of rules in $\mathcal{P}$. The initial *CBS*, at step 1, assumes the values in Δ. Initial *SPS*, *AAQ*, and *RF* are empty since there are no suspended processes, no questions were asked to the agent information sources, and no replies were received. At step 1, the first active process is created with the query as its initial goal in GS. As a consequence of several steps of *process reduction*, the process is split into new active processes with new goal sets. By step 4 the active processes represent the different alternative tasks.

At step 5, the agent needs to reduce processes with $t(T)@ois$ as the selected goal and, since there is only a default constraint for that goal in the *CBS*, that constraint is used in *process reduction*, yielding active processes which are consistent with the default and suspended processes which are not. $t(T)@ois$ is an askable atom and, as such, a question is sent to ois so as to know the real value of $t(T)@ois$. The procedure is similar whenever an askable atom is found. It is possible to observe that the active processes now have $t(T)@ois$ in *UD* and this askable atom was removed from the GS. At the same time, the suspended processes resulting from the reduction have $t(T)@ois$ as a suspended atom in the *SAS*. The results of disjunctive constraint processing are also observable; an example is the splitting of the active process concerning *action4* in step 4. At step 5 this process originates an active process and a suspended process.

By continuing *process reduction* it is possible to arrive at step 7, where there is a process with an empty goal set. This was achieved purely by relying on default constraints and, as such, the process is a scenario. By outputting *C* and *UD* at step 8, the speculative agent informs that the most likely task to follow *question1* is *action5*, which recommends a series of workup exams such as a colonoscopy, an abdominal/pelvic CT, and so forth. Based on this, the health care professional may start the preparations for executing this task. The effect of Speculative Computation with Default Revision is observed here; with no information whatsoever, it is possible to arrive at the most likely scenario in terms of probabilities, represented as an answer to the initial query. Depending on the rules in $\mathcal{P}$, there might be cases in which more than one active process with an empty goal set is produced. In such situations all the processes are presented as scenarios.

Assuming the information arrives from the ois at some point, regardless of the order, one may have:

9. $(n(N)@ois \hookleftarrow N \in \{n0\} \parallel)$ is returned from *ois* and by **Fact Arrival Phase (Fig. 5)**:

   - The returned fact replaces the default constraint in the *CBS*;
   - The returned fact is added to *RF*;

   $CBS = \{$
   $t(T)@ois \hookleftarrow T \in \{t3\} \parallel,$
   $\underline{n(N)@ois \hookleftarrow N \in \{n0\} \parallel},$
   $m(M)@ois \hookleftarrow M \in \{m1\} \parallel$
   }
   $RF = \left\{\underline{n(N)@ois \hookleftarrow N \in \{n0\} \parallel}\right\}$

   - Process P_8 in the *APS*, along with processes P_6 and P_{10} from *SPS* are removed because they are inconsistent with the returned fact;

- P_7 is resumed and originates P_{11};
- The newly activated process P_{11} still has goals in the goal set and thus $m(M)@ois$ is selected for process reduction.

$APS = \{$
$\langle\{\hookleftarrow F \in \{action5\}, T \in \{t3\}, N \in \{n0\}, M \in \{m1\} \parallel \underline{m(M)@ois}\}, \{t(T)@ois\}\rangle^{P_{11}}$
$\}$
$SPS = \{P_1, P_2, P_3, P_4, P_5, P_9\}$

10. **Process Reduction, by Case 2.2 (Fig. 4):**
Since $m(M)@ois \in AAQ$ and $((m(M)@ois \hookleftarrow M \in \{m1\} \parallel) \in \Delta)$:
$APS = \{$
$\langle\{\hookleftarrow F \in \{action5\}, T \in \{t3\}, N \in \{n0\}, M \in \{m1\} \parallel \emptyset\}, \{t(T)@ois, m(M)@ois\}\rangle^{P_{11}}$
$\}$
$SPS = \{P_1, P_2, P_3, P_4, P_5, P_9\}$

11. **By Collection of Defaults:**
Since there is now a returned fact for $n(N)@ois$, the MAP estimation is $TM_{MAP} = \max_{T,M} P(T, M \mid N = n0) \approx 0.9126$ with $T = t1$ and $M = m1$, thus producing $New\Delta = \{t(T)@ois \hookleftarrow T \in \{t1\} \parallel\}$.
As such, by **Default Revision Phase (Fig. 6):**

- The new default constraint replaces the old default constraint in the *CBS*;

$CBS = \{$
$\underline{t(T)@ois \hookleftarrow T \in \{t1\} \parallel},$
$n(N)@ois \hookleftarrow N \in \{n0\} \parallel,$
$m(M)@ois \hookleftarrow M \in \{m1\} \parallel$
$\}$

- Process P_{11} becomes suspended on $t(T)@ois$;
- Processes P_2, P_4 and P_5 are resumed, forming processes P_{12} and P_{13}, P_{14} and P_{15}, and P_{16} and P_{17}, respectively;
- Process P_9 originates process P_{18} because a new atom is added to the suspended atom set;
- The newly activated processes still have elements in the goal set, so they are further reduced according to the *CBS*.

$APS = \{$
$\langle\{\hookleftarrow F \in \{action1\}, T \in \{t1\}, N \in \{n0\}, M \in \{m0\} \parallel \underline{n(N)@ois}, m(M)@ois\}, \{t(T)@ois\}^{P_{12}},$
$\langle\{\hookleftarrow F \in \{action4\}, T \in \{t1\}, N \in \{n1, n2\}, M \in \{m0\} \parallel \underline{n(N)@ois}, m(M)@ois\}, \{t(T)@ois\}\rangle^{P_{14}},$
$\langle\hookleftarrow F \in \{action5\}, T \in \{t1\}, N \in \{n0, n1, n2\}, M \in \{m1\} \parallel \underline{n(N)@ois}, m(M)@ois\}, \{t(T)@ois\}\rangle^{P_{16}}$
$\}$
$SPS = \{$
$\langle\{\{\underline{t(T)@ois}\}, \hookleftarrow F \in \{action1\}, T \in \{t2\}, N \in \{n0\}, M \in \{m0\} \parallel n(N)@ois, m(M)@ois\}, \emptyset\rangle^{P_{13}},$
$\langle\{\{\underline{t(T)@ois}\}, \hookleftarrow F \in \{action4\}, T \in \{t2, t4\}, N \in \{n1, n2\}, M \in \{m0\} \parallel n(N)@ois, m(M)@ois\}, \emptyset\rangle^{P_{15}},$

$\langle\{\{\underline{t(T)@ois}\}, \hookleftarrow F \in \{action5\}, T \in \{t2, t4\}, N \in \{n0, n1, n2\}, M \in \{m1\} \parallel n(N)@ois, m(M)@ois\}, \emptyset\rangle^{P_{17}}$,
$\langle\{\{m(M)@ois, \underline{t(T)@ois}\}, \hookleftarrow F \in \{action2\}, T \in \{t3\}, M \in \{m0\} \parallel \emptyset\}, \emptyset\rangle^{P_{18}}$,
P_1, P_3, P_{11}
}

12. After steps of **Process Reduction, by Case 2.2 (Fig. 4):**

   - Process P_{14} is deleted because it is inconsistent with the returned fact for $n(N)@ois$;
   - Process P_{12} is suspended.

   $APS = \{$
   $\langle\{\hookleftarrow F \in \{action5\}, T \in \{t1\}, N \in \{n0\}, M \in \{m1\} \parallel \emptyset\}, \{t(T)@ois, m(M)@ois\}\rangle^{P_{16}}$

   }
   $SPS = \{$
   $\langle\{\{\underline{m(M)@ois}\}, \hookleftarrow F \in \{action1\}, T \in \{t1\}, N \in \{n0\}, M \in \{m0\} \parallel \emptyset\}, \{t(T)@ois\}\rangle^{P_{12}}$,

   $P_1, P_3, P_{11}, P_{13}, P_{15}, P_{17}, P_{18}$
   }

13. $(t(T)@ois \hookleftarrow T \in \{t1\} \parallel)$ is returned from *ois* and by **Fact Arrival Phase (Fig. 5):**

   - The returned fact coincides with the default constraint;
   - The returned fact is added to *RF*;

   CBS remains unchanged.
   $RF = \{\underline{t(T)@ois \hookleftarrow T \in \{t1\} \parallel}, n(N)@ois \hookleftarrow N \in \{n0\} \parallel\}$

   - $t(T)@ois$ is removed from *UD* in the active processes because it is no longer a default.
   - Processes P_1, P_3, P_{11}, P_{13}, P_{15}, P_{17} and P_{18} are deleted.

   $APS = \{$
   $\langle\{\hookleftarrow F \in \{action5\}, T \in \{t1\}, N \in \{n0\}, M \in \{m1\} \parallel \emptyset\}, \{m(M)@ois\}\rangle^{P_{16}}$
   }
   $SPS = \{P_{12}\}$

14. **By Collection of Defaults:**
   Since there is another returned fact, in this case for $(t(T)@ois$, the MAP estimation is $M_{MAP} = \max_M P(M \mid T = t1, N = n0) \approx 0.9126$ with $M = m1$, thus making it unnecessary to produce a $New\Delta$ and perform default revision.

15. $(m(M)@ois \hookleftarrow M \in \{m1\} \parallel)$ is returned from *ois* and by **Fact Arrival Phase (Fig. 5)**:

   - The returned fact coincides with the default constraint;
   - The returned fact is added to *RF*;

   CBS remains unchanged.
   $RF = \{$
   $t(T)@ois \hookleftarrow T \in \{t1\} \parallel$,
   $n(N)@ois \hookleftarrow N \in \{n0\} \parallel$,
   $\underline{m(M)@ois \hookleftarrow M \in \{m1\} \parallel}$
   }

   - $m(M)@ois$ is removed from *UD* in the active processes because it is no longer a default;

– Process P_{12} is deleted.

$APS = \{\langle\{\hookleftarrow F \in \{action5\}, T \in \{t1\}, N \in \{n0\}, M \in \{m1\} \parallel \emptyset\}, \emptyset\rangle^{P_{16}}\}$
$SPS = \emptyset$

16. **Process Reduction, by Case 1 (Fig. 4):**
$APS = \{$
$APS = \{\langle\{\hookleftarrow F \in \{action5\}, T \in \{t1\}, N \in \{n0\}, M \in \{m1\} \parallel \emptyset\}, \emptyset\rangle^{P_{16}}\}$
$\}$
$C = \{F \in \{action5\}, T \in \{t1\}, N \in \{n0\}, M \in \{m1\}\}$
$UD = \emptyset$
End of the execution.

In *fact arrival*, active processes may be deleted and suspended processes may be resumed as the answers from the ois are regarded as definitive. At step 9 there are two new active processes resulting from the arrival of a fact for $n(N)@ois$. The *CBS* is updated with the fact "$n(N)@ois \hookleftarrow N \in \{n0\} \parallel$", replacing the default constraint. All the processes (both active and suspended) that use the default constraint are removed from the execution, resulting in the elimination of processes P_8, P_6, and P_{10}. On the other hand, process P_7 is resumed and, after processing the constraints for $n(N)@ois$, it originates process P_{11}. In this process $n(N)@ois$ does not appear in *UD* since the askable atom is not used as a default. Furthermore, upon resuming process P_7, no alternative suspended process is generated because there cannot be another value for $n(N)@ois$ besides the newly arrived fact. Upon the arrival of the fact "$n(N)@ois \hookleftarrow N \in \{n0\} \parallel$" from the ois, the system does not add a brand new suspended process "$\langle\{\hookleftarrow F \in \{action5\}, T \in \{t3\}, N \in \{n1\}, M \in \{m1\} \parallel m(M)@ois\}, \{t(T)@ois\}\rangle$", generated from P_7, because it would not represent a feasible scenario. Further *process reduction* results in the processes of step 10.

After applying the Collection of Defaults using the newly arrived fact as evidence for the MAP estimation, a $New\Delta$ set is produced with a new default constraint for $t(T)@ois$. As such, *default revision* is performed at step 11. $t(T)@ois \hookleftarrow T \in \{t1\} \parallel$ replaces the old default constraint in the *CBS* and process P_{11} is suspended because it becomes inconsistent with the new default constraint. Previously suspended processes are resumed, yielding new active processes consistent with the new default constraint and new suspended processes consistent with the negation of the default constraint. That is what happens with processes P_2, P_4, and P_5, each one originating two new processes. Another effect of Speculative Computation with Default Revision is observed in the change operated in P_9. This process used the previous default for $t(T)@ois$ and, when revised with the new default constraint, it becomes P_{18} which is suspended on the account of two askable atoms. If this newly suspended process were to be activated it would have to be revised for both $m(M)@ois$ and $t(T)@ois$, and *SAS* would have to become an empty set. This situation can only be created with default revision. At the same time, the probability value in the MAP estimation of this step is higher than the initial MAP estimation, which indicates that the collective state of the information and, thus, the scenarios resulting from the *default revision* phase are more likely to be closer to the real outcome.

At the end of step 12, after *process reduction*, there is an active process representing the most likely scenario given the configuration of the information at the time, achieved through facts and default constraints. This is another observable effect of Speculative Computation with Default Revision; the default constraints adjusted themselves to the arrived fact, which, in turn, triggered the adjustment of processes, resulting in a scenario that is closer to reality. It is an example of how the on-line *default revision* takes place. This would inform a health care professional executing the guideline that, although the prediction changed, the most plausible answer remains the same, that is $action5$. At step 13, there is another reply from *ois*, which triggers a *fact arrival* phase, but, as the fact matches the default constraint, the change in the active process is only the removal of $t(T)@ois$ from UD. All the suspended processes that are not consistent with the newly arrived fact are removed. The MAP estimation, from the Collection of Defaults at step 14, produces the same value as in step 11 due to M being independent from the other two variables, which means that a $New\Delta$ is not produced. The remaining execution represents the arrival of the information that is still missing, until a process composed solely of facts is obtained. This happens at step 16, by outputting C and UD. The answer remains the recommendation of $action5$, achieved through a different path. The process that originated the solution, P_{16}, suffered changes, going from a process using facts and defaults to a process using only facts. This demonstrates how a speculative computation can be used in order to advance the computation of an answer and then be utilized to achieve a definitive answer.

Demonstrating all the possible cases in Speculative Computation with Default Revision with an example would be impractical. The example above only shows its main effects. However, with the claim of Sect. 3.6 it is possible to show that the procedures in the different phases of the SF_{DR} yield correct results.

It is possible to argue that simply choosing the most probable set of default constraints and deriving the most likely situation as a basis for action, with no regard for the impact that action could have, may be unrealistic or, at the very least, incomplete in terms of decision criteria.

Indeed, the framework presented herein lacks a more comprehensive and well-defined strategy for selecting not only the most plausible scenario, but also the most useful from the point of view of its costs and potential benefits. The use of probabilities, and more specifically BNs, is motivated by the notion that the knowledge we have about the world is imperfect and that, through a Bayesian approach, it is possible to get a degree of belief that something may be the case. As a complement to this degree of belief, an assessment of the strengths and weaknesses of the different alternative actions that satisfy a scenario could, or rather should, be combined with the probability resulting from the MAP assignments. This would ensure that the costs of taking a certain action do not outweigh (or far outweigh, when compared with other actions) its benefits. Cost-effectiveness analysis is used across multiple domains to appraise the desirability of an alternative, given a certain decision making problem [3]. It does so by comparing the relative costs and outcomes (effects) of two or more courses of action. Cost-effectiveness analysis is different from cost-benefit analysis in that it does not assign a monetary value to the measure of effect. It is widely used in health care for allocation decisions and to highlight interventions that are relatively inexpensive, but have the potential to reduce the disease burden substantially [18]. Taking this domain as an illustrative example, in the available literature for this kind of analysis, the costs are expressed in monetary units and the effectiveness of interventions are expressed in units of measure that are characteristic of health care such as years of life gained with an intervention or the disability-adjusted life year (which measures not only the years gained but also the improved health). By doing this, cost-effectiveness analysis, unlike cost-benefit analysis, draws attention exclusively to health benefits, which are not monetized. Another element commonly taken into account in cost-effectiveness analysis is time, more specifically the changes in costs and effects of interventions with the passage of time, which enhances the utility of an evaluation and makes it more realistic [28]. Clinical guidelines often provide references to cost-effectiveness studies regarding their recommendations. Although the guideline from which this example was retrieved does not go as far as to quantify in any way the costs and effects of the alternatives featured in the example, it features numerous references of this type [1]. The main point to retain is that the combination of MAP assignments with this kind of analysis would capture a given setting in a more realistic way and, therefore, would result in more helpful scenarios.

That being said, cost-effectiveness analysis has inherent challenges and involves a sequence of steps until a utility is achieved. Every analysis requires a host of assumptions, complex calculations, and the careful judgement of analysts. Furthermore, it is necessary to quantify the exact costs and effects of alternatives, which is not always clear and available in the knowledge about the domain. Thus, setting the framework for this kind of analysis would only be possible for very specific domains where this information is available.

6 Conclusions and future work

In this work, BNs assume a supportive role, which is slightly different from their common application. Here, the BN model itself is not the target, but an auxiliary tool. With arrived facts MAP estimation values may change, yielding new estimations for the information that is still unobserved. By dynamically revising default constraints it becomes possible to ensure that the information has the most likely configuration at all times and, in this way, produce the most likely scenarios. Speculative Computation with Default Revision is a way to manage the state of the information, fulfilling the role of an interface to rule-guided problems. The mechanism for default revision is richer and more encompassing than those presented in previous works. At the same time, it is an on-line process that reflects the dynamic nature of real environments. It takes into account the arrival of new facts to change default constraints, which is a more realistic perspective about real-world problems.

As far as we know this is the first on-line framework that uses machine learning techniques to adapt default rules. It provides a combination of logic programming-based, declarative user interface and an effective, adaptive operational behaviour in the background.

Future work includes further developing the model in order to compute utility values for the scenarios and make full use of the probability provided in the MAP estimation. So far, this probability is only used in the background, but it would be useful to include it in the different phases of speculative computation as a likelihood measure. As a complement to the MAP assignments and in order to build a utility function for speculative computation, cost-effectiveness can be explored as a strategy to depict the usefulness of alternatives more accurately and realistically. Furthermore, this kind of analysis can be made in such a way that tackles the importance of time in these decision problems. In certain cases, it may be necessary to decide before the state of a system deteriorates and changes irreversibly, which undoubtedly would affect the utility of the scenarios generated for it. This is even more evident in the domain chosen as an example for this work. In clinical decision support, health care professionals may have to make hard decisions before the state of a patient gets worse. As such, we are also working towards the inclusion of time as a relevant dimension in speculative computation.

A small prototype was developed for this example. The Generation of Defaults was implemented using the resources provided by the *bnlearn* library for R [25] and the Java *inflib* library from the SamIam project [4]. The Speculative Computation with Default Revision operational model was implemented in Prolog.

Annex 1: Table of abbreviations and symbols

See Table 1.

Table 1 Abbreviations and symbols used in the definitions and procedures of the Framework for Speculative Computation with Default Revision

Abbreviation/symbol	Meaning
Σ	Set containing the agent identifiers representing the information sources
Δ	The default answer set
APS	The set of active process processes
AAQ	The set of already asked questions
$body(R)$	The body of rule R
C_d	A default constraint
C_{newd}	A new default constraint
CBS	The current belief state
C_r	A fact constraint
$const(R)$	The constraints in the body of rule R
$\mathcal{E}$	Set containing external predicates representing askable atoms
GS	The goal set in a process
$head(R)$	The head of rule R
$New\Delta$	The set of changed default constraint rules
$NewAAQ$	An updated version of *AAQ* resulting from a process reduction phase
NewAPS	An updated version of *APS* resulting from a phase in Speculative Computation with Default Revision
NewCBS	An updated version of *CBS* resulting from an answer arrival phase or a default revision phase
NewRF	An updated version of *RF* resulting from an answer arrival phase
NewSPS	An updated version of *SPS* resulting from a phase in Speculative Computation with Default Revision
$\mathcal{P}$	The constraint logic program
$\mathcal{R}$	The reply set
RF	The set of returned facts from the agent information sources
SAS	The suspended atom set
SF_{DR}	Framework for Speculative Computation with Default Revision
SPS	The set of suspended processes
UD	The used default set of a process

Annex 2: Algorithms for speculative computation with default revision

The following algorithms specify the operations involved in the different phases of Speculative Computation with Default Revision.

Algorithms 1 and 2 refer to the initial step and iteration steps respectively of *process reduction*. In this phase, changes occur in the process sets. In the algorithms, changed *APS*, *SPS*, and *AAQ* are represented as *NewAPS*, *NewSPS* and *NewAAQ*. The last three correspond to updated versions of the previous after the step of *process reduction* is applied.

Algorithm 1 Initial step of process reduction

```
Data: $GS$: the initial goal set
Data: $\Delta$: the default answer set
give $\langle \hookleftarrow C \parallel GS, \emptyset \rangle$ to the proof procedure
$APS \leftarrow \{\langle \hookleftarrow C \parallel GS, \emptyset \rangle\}$
$SPS \leftarrow \emptyset$
$CBS \hookleftarrow \Delta$
$AAQ \leftarrow \emptyset$
$RF \leftarrow \emptyset$
```

Algorithm 2 Iteration step of process reduction

```
Data: $\Sigma$: the set of agent information sources
Data: $APS$: the set of active processes
Data: $SPS$: the set of suspended processes
Data: $CBS$: the current belief state
Data: $AAQ$: the set of already asked questions
Data: $RF$: the set of returned facts
if there is an active process $\langle \hookleftarrow C \parallel \emptyset, UD \rangle$ w.r.t. $CBS$ in $APS$ then    ▷ Case 1
    output constraints $C$ and return used defaults $UD$
else    ▷ Case 2
    select an active process $\langle \hookleftarrow C \parallel GS, UD \rangle$ from $APS$ w.r.t. $CBS$
    select an atom $L$ in $GS$
    $APS' \leftarrow APS - \{\langle \hookleftarrow C \parallel GS, UD \rangle\}$
    $GS' \leftarrow GS - \{L\}$
    for the selected atom $L$ do
        if $L$ is a non-askable atom then    ▷ Case 2.1
            $NewAPS \leftarrow APS' \cup \{\langle \hookleftarrow (C \wedge \{B_i = head(R)\} \wedge const(R) \parallel (body(R) \cup GS'), UD \rangle \mid$
                        $C \wedge \{B_i = head(R)\} \wedge const(R)$ is consistent$\}$
        else if $L$ is an askable atom $Q@S$ then    ▷ Case 2.2
            if $L \notin AAQ$ then
                send question $Q$ to agent $S$ in which $S \in \Sigma$
                $NewAAQ \leftarrow AAQ \cup \{L\}$
                $AAQ \leftarrow NewAAQ$
            end if
            if $L \in UD$ then
                $NewAPS \leftarrow APS' \cup \{\langle \hookleftarrow C \parallel GS', UD \rangle\}$
            else if $(L \hookleftarrow C_r \parallel) \in RF$ then
                if $C \wedge C_r$ is consistent then
                    $NewAPS \leftarrow APS' \cup \{\langle \hookleftarrow C \wedge C_r \parallel GS', UD \rangle\}$
                else
                    $NewAPS \leftarrow APS'$
                end if
            else if there is a default constraint $C_d$ for $L$ then
                if $C \wedge C_d$ is consistent then
                    $NewAPS \leftarrow APS' \cup \{\langle \hookleftarrow C \wedge C_d \parallel GS', UD \cup \{L\} \rangle\}$
                else
                    $NewAPS \leftarrow APS'$
                end if
                if $C \wedge \neg C_d$ is consistent then
                    $NewSPS \leftarrow SPS \cup \{\langle L, \hookleftarrow \alpha \parallel GS', UD \rangle\}$ where $C \wedge \neg C_d \models \alpha$
                    $SPS \leftarrow NewSPS$
                end if
            end if
        end if
        $APS \leftarrow NewAPS$
    end for
end if
```

Algorithm 3 describes *fact arrival*. In the algorithm, *NewRF*, *NewAPS*, *NewSPS* and *NewCBS* correspond to changed versions of *RF*, *APS*, *SPS* and *CBS* after *fact arrival*. The prefixes "*Added*—" and "*Deleted*—" indicate portions of a set that are being added or removed respectively.

Algorithm 3 Fact arrival

```
Data: $APS$: the set of active processes
Data: $SPS$: the set of suspended processes
Data: $CBS$: the current belief state
Data: $RF$: the set of returned facts
Data: $Q@S \hookleftarrow C_r \parallel$: the returned fact constraint
$NewRF \leftarrow RF \cup \{Q@S \hookleftarrow C_r \parallel\}$
$RF \leftarrow NewRF$
$NewCBS \leftarrow CBS - \{Q@S \hookleftarrow C_d \parallel\} \cup \{Q@S \hookleftarrow C_r \parallel\}$
$CBS \leftarrow NewCBS$
if $C_r$ entails $C_d$ then    ▷ Case 1
    ▷ revision of the set of active processes
    $DeletedAPS \leftarrow \{\langle \hookleftarrow \parallel GS, UD\rangle \in APS \mid Q@S \in UD\}$
    $AddedAPS \leftarrow \{\langle \hookleftarrow (C \wedge C_r) \parallel GS, UD\rangle \mid \langle \hookleftarrow C \parallel GS, UD\rangle \in DeletedAPS$ and $C \wedge C_r$
        is consistent$\}$
    $NewAPS \leftarrow APS - DeletedAPS \cup AddedAPS$
    ▷ revision of the set of suspended processes
    $DeletedSPS \leftarrow \{\langle SAS, \hookleftarrow C \parallel GS, UD\rangle \in SPS \mid Q@S \in SAS$ or $Q@S \in UD\}$
    $AddedSPS \leftarrow \{\langle SAS, \hookleftarrow (C \wedge C_r) \parallel GS, UD\rangle \mid \langle SAS, \hookleftarrow C \parallel GS, UD\rangle$
        $\in DeletedSPS$, $Q@S \in UD$, and $C \wedge C_r$ is consistent$\}$
    $NewSPS \leftarrow SPS - DeletedSPS \cup AddedSPS$
else if $C_r$ contradicts $C_d$ then    ▷ Case 2
    ▷ revision of the set of active processes
    $DeletedAPS \leftarrow \{\langle \hookleftarrow C \parallel GS, UD\rangle \in APS \mid Q@S \in UD\}$
    $ResumedSPS \leftarrow \{\langle \hookleftarrow (C \wedge C_r) \parallel GS, UD\rangle \mid \langle \{Q@S\}, \hookleftarrow C \parallel GS, UD\rangle \in SPS$, and $C \wedge C_r$
        is consistent$\}$
    $NewAPS \leftarrow APS - DeletedAPS \cup ResumedSPS$
    ▷ revision of the set of suspended processes
    $DeletedSPS \leftarrow \{\langle SAS, \hookleftarrow C \parallel GS, UD\rangle \in SPS \mid Q@S \in SAS$ or $Q@S \in UD\}$
    $AddedSPS \leftarrow \{\langle SAS - \{Q@S\}, \hookleftarrow (C \wedge C_r) \parallel GS, UD \cup \{Q@S\}\rangle \mid \langle SAS, \hookleftarrow C \parallel GS, UD\rangle \in$
        $DeletedSPS$, $Q@S \in SAS$, $SAS - \{Q@S\} \neq \emptyset$, and $C \wedge C_r$ is consistent$\}$
    $NewSPS \leftarrow SPS - DeletedSPS \cup AddedSPS$
else if $C_r$ does not entail $C_d$ or contradict $C_d$ then    ▷ Case 3
    ▷ revision of the set of active processes
    $DeletedAPS \leftarrow \{\langle \hookleftarrow C \parallel GS, UD\rangle \in APS \mid Q@S \in UD\}$
    $AddedAPS \leftarrow \{\langle \hookleftarrow (C \wedge C_r) \parallel GS, UD\rangle \mid \langle \hookleftarrow C \parallel GS, UD\rangle \in DeletedAPS$ and $C \wedge C_r$
        is consistent$\}$
    $ResumedSPS \leftarrow \{\langle \hookleftarrow (C \wedge C_r) \parallel GS, UD\rangle \mid \langle \{Q@S\}, \hookleftarrow C \parallel GS, UD\rangle$
        $\in SPS$ and $C \wedge C_r$ is consistent$\}$
    $NewAPS \leftarrow APS - DeletedAPS \cup AddedAPS \cup ResumedSPS$
    ▷ revision of the set of suspended processes
    $DeletedSPS \leftarrow \{\langle SAS, \hookleftarrow (C \wedge C_r) \parallel GS, UD\rangle \in SPS \mid Q@S \in SAS$ or $Q@S \in UD\}$
    $AddedSPS1 \leftarrow \{\langle SAS, \hookleftarrow (C \wedge C_r) \parallel GS, UD\rangle \mid \langle SAS, \hookleftarrow C \parallel GS, UD\rangle$
        $\in DeletedSPS$, $Q@S \in UD$, and $C \wedge C_r$ is consistent$\}$
    $AddedSPS2 \leftarrow \{\langle SAS - \{Q@S\}, \hookleftarrow (C \wedge C_r) \parallel GS, UD \cup \{Q@S\}\rangle \mid \langle SAS, \hookleftarrow C \parallel GS, UD\rangle \in$
        $DeletedSPS$, $Q@S \in SAS$, $SAS - \{Q@S\} \neq \emptyset$, and $C \wedge C_r$ is consistent$\}$.
    $NewSPS \leftarrow SPS - DeletedSPS \cup AddedSPS1 \cup AddedSPS2$
end if
$APS \leftarrow NewAPS$
$SPS \leftarrow NewSPS$
```

Finally, Algorithms 4 and 5 describe *default revision*. In the procedures given in the algorithms, *NewAPS*, *NewSPS*, and *NewCBS* correspond to changed versions of *APS*, *SPS* and *CBS*. The prefixes "*Added*−" and "*Deleted*−" indicate portions of a set that are being added or removed respectively.

Algorithm 4 Default revision

```
Data: $APS$: the set of active processes
Data: $SPS$: the set of suspended processes
Data: $CBS$: the current belief state
Data: $New\Delta$: the set of changed default constraints
Data: $Q_d@S \hookleftarrow C_{newd} \,\|$: the new default constraint
if $New\Delta \neq \emptyset$ then
  for each $Q_d@S \hookleftarrow C_{newd} \,\|. \in new\Delta$ do
    $NewCBS \leftarrow CBS - \{Q_d@S \hookleftarrow C_d \,\|\} \cup \{Q_d@S \hookleftarrow C_{newd} \,\|\}$
    if $C_{newd}$ entails $C_d$ then                                        ▷ Case 1
                                        ▷ revision of the set of active processes
      $DeletedAPS \leftarrow \{\langle \hookleftarrow C \,\|\, GS, UD\rangle \in APS \mid Q_d@S \in UD\}$
      $AddedAPS \leftarrow \{\langle \hookleftarrow (C \wedge C_{newd}) \,\|\, GS, UD\rangle \mid \langle \hookleftarrow C \,\|\, GS, UD\rangle \in DeletedAPS$ and
                  $C \wedge C_{newd}$ is consistent$\}$
      $NewAPS \leftarrow APS - DeletedAPS \cup AddedAPS$
                                        ▷ revision of the set of suspended processes
      $DeletedSPS \leftarrow \{\langle SAS, \hookleftarrow C \,\|\, GS, UD\rangle \in SPS \mid Q_d@S \in UD\}$
      $AddedSPS1 \leftarrow \{\langle SAS, \hookleftarrow (C \wedge C_{newd}) \,\|\, GS, UD\rangle \mid \langle SAS, \hookleftarrow C \,\|\, GS, UD\rangle \in$
                  $DeletedSPS$, $Q_d@S \in UD$, and $C \wedge C_{newd}$ is consistent$\}$
      $AddedSPS2 \leftarrow \{\langle SAS \cup \{Q_d@S\}, \hookleftarrow C \,\|\, GS, UD - \{Q_d@S\}\rangle \mid \langle SAS, \hookleftarrow C \,\|\, GS, UD\rangle$
                  $\in DeletedSPS$ and $Q_d@S \in UD\}$
      $SuspendedAPS \leftarrow \{\langle \{Q_d@S\}, \hookleftarrow (C \wedge \neg C_{newd}) \,\|\, GS, UD\rangle \mid \langle \hookleftarrow C \,\|\, GS, UD\rangle \in APS$
                  and $C \wedge \neg C_{newd}$ is consistent$\}$
      $NewSPS \leftarrow SPS - DeletedSPS \cup AddedSPS1 \cup AddedSPS2 \cup AddedSPS3$
                  $\cup\, SuspendedAPS$
    else if $C_{newd}$ contradicts $C_d$ then                                ▷ Case 2
                                        ▷ revision of the set of active processes
      $DeletedAPS \leftarrow \{\langle \hookleftarrow C \,\|\, GS, UD\rangle \in APS \mid Q_d@S \in UD\}$
      $ResumedSPS \leftarrow \{\langle \hookleftarrow (C \wedge C_{newd}) \,\|\, GS, UD\rangle \mid \langle \{Q_d@S\}, \hookleftarrow C \,\|\, GS, UD\rangle \in SPS$ and
                  $C \wedge C_{newd}$ is consistent$\}$
      $NewAPS \leftarrow APS - DeletedAPS \cup ResumedSPS$
                                        ▷ revision of the set of suspended processes
      $DeletedSPS \leftarrow \{\langle SAS, \hookleftarrow C \,\|\, GS, UD\rangle \in SPS \mid Q_d@S \in SAS$ or $Q_d@S \in UD\}$
      $AddedSPS1 \leftarrow \{\langle SAS \cup \{Q_d@S\}, \hookleftarrow (C \wedge \neg C_{newd}) \,\|\, GS, UD - \{Q_d@S\}\rangle \mid \langle SAS, \hookleftarrow C$
                  $\|\, GS, UD\rangle \in DeletedSPS$, $Q_d@S \in UD$ and $C \wedge \neg C_{newd}$ is consistent$\}$
      $AddedSPS2 \leftarrow \{\langle SAS \cup \{Q_d@S\}, \hookleftarrow C \,\|\, GS, UD - \{Q_d@S\}\rangle \mid \langle SAS, \hookleftarrow C \,\|\, GS, UD\rangle$
                  $\in DeletedSPS$ and $Q_d@S \in UD\}$
      $AddedSPS3 \leftarrow \{\langle SAS - \{Q_d@S\}, \hookleftarrow (C \wedge C_{newd}) \,\|\, GS, UD \cup \{Q_d@S\}\rangle \mid \langle SAS, \hookleftarrow C \,\|$
                  $GS, UD\rangle \in DeletedSPS$, $Q_d@S \in SAS$, $SAS - \{Q_d@S\} \neq \emptyset$, and
                  $C \wedge C_{newd}$ is consistent$\}$
      $SuspendedAPS \leftarrow \{\langle \{Q_d@S\}, \hookleftarrow C \,\|\, GS, UD\rangle \mid \langle \hookleftarrow C \,\|\, GS, UD\rangle \in APS$ and
                  $Q_d@S \in UD\}$
      $NewSPS \leftarrow SPS - DeletedSPS \cup AddedSPS1 \cup AddedSPS2 \cup AddedSPS3$
                  $\cup\, SuspendedAPS$
```

Algorithm 5 Default revision - continued

```
    else if $C_{newd}$ does not entail $C_d$ or contradict $C_d$ then                ▷ Case 3
                                              ▷ revision of the set of active processes
        $DeletedAPS \leftarrow \{\langle \hookleftarrow C \parallel GS, UD\rangle \in APS \mid Q_d@S \in UD\}$
        $AddedAPS \leftarrow \{\langle \hookleftarrow (C \wedge C_{newd}) \parallel GS, UD\rangle \mid \langle \hookleftarrow C \parallel GS, UD\rangle \in DeletedAPS$ and
                $C \wedge C_{newd}$ is consistent$\}$
        $ResumedSPS \leftarrow \{\langle \hookleftarrow (C \wedge C_{newd}) \parallel GS, UD\rangle \mid \langle \{Q_d@S\}, \hookleftarrow C \parallel GS, UD\rangle \in SPS$ and
                $C \wedge C_{newd}$ is consistent$\}$
        $NewAPS \leftarrow APS - DeletedAPS \cup AddedAPS \cup ResumedSPS$
                                           ▷ revision of the set of suspended processes
        $DeletedSPS \leftarrow \{\langle SAS, \hookleftarrow C \parallel GS, UD\rangle \in SPS \mid Q_d@S \in SAS$ or $Q_d@S \in UD\}$
        $AddedSPS1 \leftarrow \{\langle SAS, \hookleftarrow (C \wedge C_{newd}) \parallel GS, UD\rangle \mid \langle SAS, \hookleftarrow C \parallel GS, UD\rangle \in$
                $DeletedSPS$, $Q_d@S \in UD$, and $C \wedge C_{newd}$ is consistent$\}$
        $AddedSPS2 \leftarrow \{\langle SAS \cup \{Q_d@S\}, \hookleftarrow (C \wedge \neg C_{newd}) \parallel GS, UD - \{Q_d@S\}\rangle \mid \langle SAS, \hookleftarrow C$
                $\parallel GS, UD\rangle \in DeletedSPS$, $Q_d@S \in UD$, and $C \wedge \neg C_{newd}$ is consistent$\}$
        $AddedSPS3 \leftarrow \{\langle SAS, \hookleftarrow (C \wedge \neg C_{newd}) \parallel GS, UD\rangle \mid \langle SAS, \hookleftarrow C \parallel GS, UD\rangle \in$
                $DeletedSPS$, $Q_d@S \in SAS$, and $C \wedge \neg C_{newd}$ is consistent$\}$
        $AddedSPS4 \leftarrow \{\langle SAS - \{Q_d@S\}, \hookleftarrow (C \wedge C_{newd}) \parallel GS, UD \cup \{Q_d@S\}\rangle \mid \langle SAS, \hookleftarrow C \parallel$
                $GS, UD\rangle \in DeletedSPS$, $Q_d@S \in SAS$, $SAS - \{Q_d@S\} \neq \emptyset$, and
                $C \wedge C_{newd}$ is consistent$\}$
        $SuspendedAPS \leftarrow \{\langle \{Q_d@S\}, \hookleftarrow (C \wedge \neg C_{newd}) \parallel GS, UD\rangle \mid \langle \hookleftarrow C \parallel GS, UD\rangle \in APS$
                and $C \wedge \neg C_{newd}$ is consistent$\}$
        $NewSPS \leftarrow SPS - DeletedSPS \cup AddedSPS1 \cup AddedSPS2 \cup AddedSPS3 \cup$
                $AddedSPS4 \cup SuspendedAPS$
    end if
    $CBS \leftarrow NewCBS$
    $APS \leftarrow NewAPS$
    $SPS \leftarrow NewSPS$
  end for
end if
```

Annex 3: Sketch of proof for Claim 1

In the following sketch of proof, a round is defined as the execution of operations in the iteration step part of *process reduction*, the whole of *fact arrival*, or the whole of *default revision*, from their beginning to their end, without returning to the beginning and without transferring to another phase.

It is shown that a more general property holds for active or suspended processes at each round. The property, represented as λ, is the following: at any kth round, for any active process or suspended process $\mathcal{P}$ without negation of constraints, there exists a sequence of reductions

“$\hookleftarrow\parallel GS_{init}$”, ..., “$\hookleftarrow C'_P \parallel GS_P$”
w.r.t. $\mathcal{P}$ and
$\mathcal{R}_P^{(k)} = RF_k \cup \{\delta_k(Q@S) | Q@S \in UD_P\}$
s.t. $\pi_V(C_P)$ entails $\pi_V(C'_P)$,
where $C_P = C_{P_a}$, $GS_P = GS_{P_a}$, and $UD_P = UD_{P_a}$ if P is an active process $P_a = \langle \hookleftarrow C_{P_a} \parallel GS_{P_a}, UD_{P_a}\rangle$, and $C_P = C_{P_s}$, $GS_P = SAS_{P_s} \cup GS_{P_s}$, and $UD_P = UD_{P_s}$, if P is a suspended process $P_s = \langle SAS_{P_s} \hookleftarrow C_{P_s} \parallel GS_{P_s}, UD_{P_s}\rangle$, and RF_k is the reply set of constraints returned at the kth round.

Property λ is proven by mathematical induction for the progress of the three phases of Speculative Computation with Default Revision: *process reduction*,*fact arrival*, and *default revision*. The induction base and the induction step are the following:

- *Induction base* For $k = 0$, which is the initial round of Speculative Computation with Default Revision and corresponds to the initial step of *process reduction*, an active process $\langle \hookleftarrow \| \ GS_{init}, \emptyset \rangle$ is created, which satisfies property λ.
- *Induction step* Assume that at the kth round property λ holds. Now, considering the $(k+1)$th round, it is shown that λ holds in the following way:

1. In the case that the round is *a step of process reduction:* It is straightforward to show that λ holds for a step of *process reduction* because no constraint is added to the reply set from the previous round. It is the normal reduction of a process according to the operational model.
2. In the case that the round is a *fact arrival*:
   In this phase, consider the processing of a returned answer $Q@S \hookleftarrow C_r$ in the *fact arrival* phase and the existence of a default constraint C_d for $Q@S$.
   Let $P_a = \langle \hookleftarrow C_{P_a} \ \| \ GS_{P_a}, UD_{P_a} \rangle$ be any existing active process s.t. $Q@S \in UD_{P_a}$. In this round, P_a is deleted. For the newly added active process created from P_a one can have the following three cases:
   (a) In the case that C_r entails C_d. If $C_{P_a} \wedge C_r$ is consistent, the active process $P'_a = \langle \hookleftarrow C_{P_a} \wedge C_r \ \| \ GS_{P_a}, UD_{P_a} - \{Q@S\} \rangle$ is created and one has $\mathcal{R}^{(k+1)}_{P'_a} = R^{(k)}_{P_a} \cup \{Q@S \hookleftarrow C_r \ \|\} \backslash \{Q@S \hookleftarrow C_d \ \|\}$. By the induction hypothesis, P_a satisfies λ for some C'_{P_a}, i.e, there exists a sequence of reductions "$\hookleftarrow \| \ GS_{init}$", ..., "$\hookleftarrow C_1 \ \| \ \{Q@S\} \cup GS$", "$\hookleftarrow C_1 \wedge C_d \ \| \ GS$", ..., "$\hookleftarrow C_1 \wedge C_d \wedge C_2 \ \| \ GS_{P_a}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k)}_{P_a}$ s.t. $\pi_V(C_{P_a})$ entails $\pi_V(C_1 \wedge C_d \wedge C_2)$, where C_1 and C_2 are the constraints obtained before and after processing $Q@S$ respectively, and $C_1 \wedge C_d \wedge C_2 = C'_{P_a}$. Then one can consider the sequence of reductions "$\hookleftarrow \| \ GS_{init}$", ..., "$\hookleftarrow C_1 \ \| \ \{Q@S\} \cup GS$", "$\hookleftarrow C_1 \wedge C_r \ \| \ GS$", ..., "$\hookleftarrow C_1 \wedge C_r \wedge C_2 \ \| \ GS_{P_a}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_a}$. Since C_r entails C_d and $\pi_V(C_{P_a})$ entails $\pi_V(C_1 \wedge C_d \wedge C_2)$, then $\pi_V(C_{P_a} \wedge C_r)$ entails $\pi_V(C_1 \wedge C_r \wedge C_2)$. Thus λ holds for this new process.
   (b) In the case that C_r contradicts C_d. No active process is created from P_a.
   (c) In the case that C_r does not entail or contradict C_d. If $C_{P_a} \wedge C_r$ is consistent, an active process is created from P_a, for which one can show λ in a similar way to case (a).
   Now, let $P_s = \langle SAS_{P_s} \hookleftarrow C_{P_s} \ \| \ GS_{P_s}, UD_{P_s} \rangle$ be an existing suspended process s.t. $Q@S \in SAS_{P_s}$ or $Q@S \in UD_{P_s}$. P_s is deleted at this step.
   Consider the newly added active process that is created from P_s s.t. $Q@S \in SAS_{P_s}$ and $SAS_{P_s} - \{Q@S\} = \emptyset$ (which corresponds to the resumed process). One can have the following cases:
   (a) In the case that C_r entails C_d. No active process is created from P_s.
   (b) In the case that C_r contradicts C_d. If $C_{P_s} \wedge C_r$ is consistent, the active process $P'_a = \langle \hookleftarrow C_{P_s} \wedge C_r \ \| \ GS_{P_s}, UD_{P_s} \cup \{Q@S\} \rangle$ is created and one has $\mathcal{R}^{(k+1)}_{P'_a} = R^{(k)}_{P_s} \cup \{Q@S \hookleftarrow C_r \ \|\} \backslash \{Q@S \hookleftarrow C_d \ \|\}$. By the induction hypothesis, P_s satisfies λ for some C'_{P_s}, i.e, there exists a sequence of reductions "$\hookleftarrow \| \ GS_{init}$", ..., "$\hookleftarrow C'_{P_s} \ \| \ \{Q@S\} \cup GS_{P_s}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_a}$ s.t. $\pi_V(C_{P_s})$ entails $\pi_V(C'_{P_s})$. Then one can consider the sequence of reductions "$\hookleftarrow \| \ GS_{init}$", ..., "$\hookleftarrow C'_{P_s} \ \| \ \{Q@S\} \cup GS$", "$\hookleftarrow C'_{P_s} \wedge C_r \ \| \ GS_{P_s}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_a}$. Since $\pi_V(C_{P_s})$

entails $\pi_V(C'_{P_s})$, then $\pi_V(C_{P_s} \wedge C_r)$ entails $\pi_V(C'_{P_s} \wedge C_r)$. Thus λ holds for this new process.

(c) In the case that C_r does not entail or contradict C_d. If $C_{P_s} \wedge C_r$ is consistent, an active process is created from P_s, for which one can show λ in a similar way to case (b).

Consider the newly added suspended process that is created from P_s so that $Q@S \in SAS_{P_s}$ and $SAS_{P_s} - \{Q@S\} \neq \emptyset$. One can have the following cases:

(a) In the case that C_r entails C_d. No new suspended process is created from P_s.

(b) In the case that C_r contradicts C_d. If $C_{P_s} \wedge C_r$ is consistent, the suspended process $P'_s = \langle SAS_{P_s} - \{Q@S\}, \hookleftarrow C_{P_s} \wedge C_r \parallel GS_{P_s}, UD_{P_s} \cup \{Q@S\}\rangle$ is created and one has $\mathcal{R}^{(k+1)}_{P'_s} = R^{(k)}_{P_s} \cup \{Q@S \hookleftarrow C_r \parallel\} \backslash \{Q@S \hookleftarrow C_d \parallel\}$. By the induction hypothesis, P_s satisfies λ for some C'_{P_s}, i.e, there exists a sequence of reductions "$\hookleftarrow\parallel GS_{init}$", …, "$\hookleftarrow C'_{P_s} \parallel \{Q@S\} \cup GS_{P_s}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_s}$ so that $\pi_V(C_{P_s})$ entails $\pi_V(C'_{P_s})$. Then one can consider the sequence of reductions "$\hookleftarrow\parallel GS_{init}$", …, "$\hookleftarrow C'_{P_s} \parallel \{Q@S\} \cup GS$", "$\hookleftarrow C'_{P_s} \wedge C_r \parallel GS_{P_s}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_s}$. Since $\pi_V(C_{P_s})$ entails $\pi_V(C'_{P_s})$, then $\pi_V(C_{P_s} \wedge C_r)$ entails $\pi_V(C'_{P_s} \wedge C_r)$. Thus λ holds for this new process.

(c) In the case that C_r does not entail or contradict C_d. If $C_{P_s} \wedge C_r$ is consistent, a suspended process is created from P_s, for which one can show λ in a similar way to case (b).

Now, consider the newly added suspended process that is created from P_s s.t. $Q@S \in UD_{P_s}$. One may have the following three cases:

(a) In the case that C_r entails C_d. If $C_{P_s} \wedge C_r$ is consistent, the suspended process $P'_s = \langle SAS_{P_s} \hookleftarrow C_{P_s} \wedge C_r \parallel GS_{P_s}, UD_{P_s}\rangle$ is created and one has $\mathcal{R}^{(k+1)}_{P'_s} = R^{(k)}_{P_s} \cup \{Q@S \hookleftarrow C_r \parallel\} \backslash \{Q@S \hookleftarrow C_d \parallel\}$. By the induction hypothesis, P_s satisfies λ for some C'_{P_s}, i.e, there exists a sequence of reductions "$\hookleftarrow\parallel GS_{init}$", …, "$\hookleftarrow C_1 \parallel \{Q@S\} \cup GS$", "$\hookleftarrow C_1 \wedge C_d \parallel GS$", …, "$\hookleftarrow C_1 \wedge C_d \wedge C_2 \parallel GS_{P_s}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k)}_{P_s}$ so that $\pi_V(C_{P_s})$ entails $\pi_V(C_1 \wedge C_d \wedge C_2)$, where C_1 and C_2 are the constraints obtained before and after processing $Q@S$ respectively, and $C_1 \wedge C_d \wedge C_2 = C'_{P_s}$. Then one can consider the sequence of reductions $\langle\hookleftarrow\parallel GS_{init}\rangle$, …, $\langle\hookleftarrow C_1 \parallel \{Q@S\} \cup GS\rangle$, $\langle\hookleftarrow C_1 \wedge C_r \parallel GS\rangle$, …, $\langle\hookleftarrow C_1 \wedge C_r \wedge C_2 \parallel \{SAS_{P_s}\} \cup GS_{P_s}\rangle$ w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_s}$. Since C_r entails C_d and $\pi_V(C_{P_s})$ entails $\pi_V(C_1 \wedge C_d \wedge C_2)$, then $\pi_V(C_{P_s} \wedge C_r)$ entails $\pi_V(C_1 \wedge C_r \wedge C_2)$. Thus λ holds for this new process.

(b) In the case that C_r contradicts C_d. No suspended process is created from P_s.

(c) In the case that C_r does not entail or contradict C_d. If $C_{P_s} \wedge C_r$ is consistent, a suspended process is created from P_s, for which one can show λ in a similar way to case 1.

As a result of *fact arrival*, no suspended process is created from an active process since the constraints from the replies of the agent information sources are regarded as definitive.

3. In the case that the round is a *default revision*:

Let $P_a = \langle\hookleftarrow C_{P_a} \parallel GS_{P_a}, UD_{P_a}\rangle$ be an existing active process so that $Q@S \in UD_{P_a}$. In this step P_a is deleted. For the newly added active process created from P_a one can have the following three cases:

(a) In the case that C_{newd} entails C_d. If $C_{P_a} \wedge C_{newd}$ is consistent, the active process $P'_a = \langle \hookleftarrow C_{P_a} \wedge C_{newd} \parallel GS_{P_a}, UD_{P_a} \rangle$ is created and one has $\mathcal{R}^{(k+1)}_{P'_a} = R^{(k)}_{P_a} \cup \{Q@S \hookleftarrow C_{newd} \parallel\} \backslash \{Q@S \hookleftarrow C_d \parallel\}$. By the induction hypothesis, P_a satisfies λ for some C'_{P_a}, i.e, there exists a sequence of reductions "$\hookleftarrow \parallel GS_{init}$", ..., "$\hookleftarrow C_1 \parallel \{Q@S\} \cup GS$", "$\hookleftarrow C_1 \wedge C_d \parallel GS$", ..., "$\hookleftarrow C_1 \wedge C_d \wedge C_2 \parallel GS_{P_a}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k)}_{P_a}$ so that $\pi_V(C_{P_a})$ entails $\pi_V(C_1 \wedge C_d \wedge C_2)$, where C_1 and C_2 are the constraints obtained before and after processing $Q@S$ respectively, and $C_1 \wedge C_d \wedge C_2 = C'_{P_a}$. Then one can consider the sequence of reductions "$\hookleftarrow \parallel GS_{init}$", ..., "$\hookleftarrow C_1 \parallel \{Q@S\} \cup GS$", "$\hookleftarrow C_1 \wedge C_{newd} \parallel GS$", ..., "$\hookleftarrow C_1 \wedge C_{newd} \wedge C_2 \parallel GS_{P_a}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_a}$. Since C_{newd} entails C_d and $\pi_V(C_{P_a})$ entails $\pi_V(C_1 \wedge C_{newd} \wedge C_2)$, $\pi_V(C_{P_a} \wedge C_d)$ entails $\pi_V(C_1 \wedge C_{newd} \wedge C_2)$. Thus λ holds for this new process.

(b) In the case that C_{newd} contradicts C_d. No active process is created from P_a.

(c) In the case that C_{newd} does not entail or contradict C_d. If $C_{P_a} \wedge C_{newd}$ is consistent, an active process is created from P_a, for which one can show λ in a similar way to case (a).

Now, let $P_s = \langle SAS_{P_s} \hookleftarrow C_{P_s} \parallel GS_{P_s}, UD_{P_s} \rangle$ be an existing suspended process so that $Q@S \in SAS_{P_s}$ or $Q@S \in UD_{P_s}$. P_s is deleted at this step.

Consider the newly added active process that is created from P_s so that $Q@S \in SAS_{P_s}$ and $SAS_{P_s} - \{Q@S\} = \emptyset$ (which corresponds to the resumed process). One can have the following cases:

(a) In the case that C_{newd} entails C_d. No active process is created from P_s.

(b) In the case that C_{newd} contradicts C_d. If $C_{P_s} \wedge C_{newd}$ is consistent, the active process $P'_a = \langle \hookleftarrow C_{P_s} \wedge C_{newd} \parallel GS_{P_s}, UD_{P_s} \cup \{Q@S\} \rangle$ is created and one has $\mathcal{R}^{(k+1)}_{P'_a} = R^{(k)}_{P_s} \cup \{Q@S \hookleftarrow C_{newd} \parallel\} \backslash \{Q@S \hookleftarrow C_d \parallel\}$. By the induction hypothesis, P_s satisfies λ for some C'_{P_s}, i.e, there exists a sequence of reductions "$\hookleftarrow \parallel GS_{init}$", ..., "$\hookleftarrow C'_{P_s} \parallel \{Q@S\} \cup GS_{P_s}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_s}$ so that $\pi_V(C_{P_s})$ entails $\pi_V(C'_{P_s})$. Then one can consider the sequence of reductions "$\hookleftarrow \parallel GS_{init}$", ..., "$\hookleftarrow C'_{P_s} \parallel \{Q@S\} \cup GS$", "$\hookleftarrow C'_{P_s} \wedge C_{newd} \parallel GS_{P_s}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_a}$. Since $\pi_V(C_{P_s})$ entails $\pi_V(C'_{P_s})$, $\pi_V(C_{P_s} \wedge C_{newd})$ entails $\pi_V(C'_{P_s} \wedge C_{newd})$. Thus λ holds for this new process.

(c) In the case that C_{newd} does not entail or contradict C_d. If $C_{P_s} \wedge C_{newd}$ is consistent, an active process is created from P_s, for which one can show λ in a similar way to case (b).

Consider the newly added suspended process that is created from P_s so that $Q@S \in SAS_{P_s}$ and $SAS_{P_s} - \{Q@S\} \neq \emptyset$. One can have the following cases:

(a) In the case that C_{newd} entails C_d. No new suspended process is created from P_s.

(b) In the case that C_{newd} contradicts C_d. If $C_{P_s} \wedge C_{newd}$ is consistent, the suspended process $P'_s = \langle SAS_{P_s} - \{Q@S\}, \hookleftarrow C_{P_s} \wedge C_{newd} \parallel GS_{P_s}, UD_{P_s} \cup \{Q@S\} \rangle$ is created and one has $\mathcal{R}^{(k+1)}_{P'_s} = R^{(k)}_{P_s} \cup \{Q@S \hookleftarrow C_{newd} \parallel\} \backslash \{Q@S \hookleftarrow C_d \parallel\}$. By the induction hypothesis, P_s satisfies λ for some C'_{P_s}, i.e, there exists a sequence of reductions "$\hookleftarrow \parallel GS_{init}$", ..., "$\hookleftarrow C'_{P_s} \parallel \{Q@S\} \cup GS_{P_s}$" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_s}$ so that $\pi_V(C_{P_s})$ entails $\pi_V(C'_{P_s})$. Then one can consider the sequence of reductions "$\hookleftarrow \parallel GS_{init}$", ..., "$\hookleftarrow C'_{P_s} \parallel \{Q@S\} \cup GS$", "$\hookleftarrow C'_{P_s} \wedge C_{newd} \parallel$

GS_{P_s}" w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_s}$. Since $\pi_V(C_{P_s})$ entails $\pi_V(C'_{P_s})$, $\pi_V(C_{P_s} \wedge C_{newd})$ entails $\pi_V(C'_{P_s} \wedge C_{newd})$. Thus λ holds for this new process.

(c) In the case that C_{newd} does not entail or contradict C_d. If $C_{P_s} \wedge C_{newd}$ is consistent, a suspended process is created from P_s, for which one can show λ in a similar way to case (b).

Now, consider the newly added suspended process that is created from P_s so that $Q@S \in UD_{P_s}$. One may have the following three cases.

(a) In the case that C_{newd} entails C_d. If $C_{P_s} \wedge C_{newd}$ is consistent, the suspended process $P'_s = \langle SAS_{P_s} \hookleftarrow C_{P_s} \wedge C_{newd} \parallel GS_{P_s}, UD_{P_s}\rangle$ is created and one has $\mathcal{R}^{(k+1)}_{P'_s} = R^{(k)}_{P_s} \cup \{Q@S \hookleftarrow C_{newd} \parallel\} \backslash \{Q@S \hookleftarrow C_d \parallel\}$. By the induction hypothesis, P_s satisfies λ for some C'_{P_s}, i.e, there exists a sequence of reductions $\langle \hookleftarrow \parallel GS_{init}\rangle, \ldots, \langle \hookleftarrow C_1 \parallel \{Q@S\} \cup GS\rangle, \langle \hookleftarrow C_1 \wedge C_d \parallel GS\rangle, \ldots, \langle \hookleftarrow C_1 \wedge C_d \wedge C_2 \parallel GS_{P_s}\rangle$ w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k)}_{P_s}$ so that $\pi_V(C_{P_s})$ entails $\pi_V(C_1 \wedge C_d \wedge C_2)$, where C_1 and C_2 are the constraints obtained before and after processing $Q@S$ respectively, and $C_1 \wedge C_d \wedge C_2 = C'_{P_s}$. Then one can consider the sequence of reductions $\langle \hookleftarrow \parallel GS_{init}\rangle, \ldots, \langle \hookleftarrow C_1 \parallel \{Q@S\} \cup GS\rangle, \langle \hookleftarrow C_1 \wedge C_{newd} \parallel GS\rangle, \ldots, \langle \hookleftarrow C_1 \wedge C_{newd} \wedge C_2 \parallel \{SAS_{P_s}\} \cup GS_{P_s}\rangle$ w.r.t. $\mathcal{P}$ and $\mathcal{R}^{(k+1)}_{P'_s}$. Since C_{newd} entails C_d and $\pi_V(C_{P_s})$ entails $\pi_V(C_1 \wedge C_d \wedge C_2)$, then $\pi_V(C_{P_s} \wedge C_{newd})$ entails $\pi_V(C_1 \wedge C_{newd} \wedge C_2)$. Thus λ holds for this new process.

(b) In the case that C_{newd} contradicts C_d. No suspended process is created from P_s.

(c) In the case that C_{newd} does not entail or contradict C_d. If $C_{P_s} \wedge C_{newd}$ is consistent, a suspended process is created from P_s, for which one can show λ in a similar way to case (a).

Processes that fall outside the scope of this proof still hold property λ since $Q@S$ has not been used in their reduction. As such, property λ holds for any active or suspended process without negation of constraints after the *fact arrival* and *default revision* phases.

References

1. Benson, A., Bekaii-Saab, T., Chan, E., Chen, Y. J., Choti, M., Cooper, H., et al. (2013). *NCCN clinical practice guideline in oncology colon cancer*. Technical Report. National Comprehensive Cancer Network. http://www.nccn.org/professionals/physician_gls/f_guidelines.asp.
2. Burton, F. W. (1985). Speculative computation, parallelism and functional programming. *IEEE Transactions on Computers*, *34*(12), 1190–1193.
3. Cellini, S. R., & Kee, J. E. (2010). Cost effectiveness and cost-benefit analysis. In *Handbook of practical program evaluation* (pp. 493–530). Wiley Online Library. doi:10.1002/9781119171386.ch24.
4. Darwiche, A. (2010). Samiam—Sensitivity analysis, modelling, inference and more. Accessed from March 15, 2015, from http://reasoning.cs.ucla.edu/samiam/.
5. Franzén, T., Haridi, S., & Janson, S. (1992). An overview of the andorra kernel language. In L. H. Eriksson, L. HallnÃd's, & P. Schroeder-Heister (Eds.), *Extensions of logic programming* (Vol. 596, pp. 163–179). Lecture notes in computer science. Berlin: Springer.
6. Governatori, G., Olivieri, F., Scannapieco, S., & Cristani, M. (2010). Superiority based revision of defeasible theories. In M. Dean, D. J. Hall, A. Rotolo, & S. Tabet (Eds.), *Semantic web rules* (Vol. 6403, pp. 104–118). Lecture notes in computer science. Berlin: Springer.
7. Gupta, V., Jagadeesan, R., & Saraswat, V. (1997). Probabilistic concurrent constraint programming. In A. Mazurkiewicz & J. Winkowski (Eds.), *CONCUR'97: Concurrency theory* (Vol. 1243, pp. 243–257). Lecture notes in computer science. Berlin: Springer.
8. Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., & Tibshirani, R. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York: Springer.
9. Hosobe, H., Satoh, K., & Codognet, P. (2007). Agent-based speculative constraint processing. *IEICE Transactions on Information and Systems, E90-D*(9), 1354–1362. doi:10.1093/ietisy/e90-d.9.1354

10. Isern, D., & Moreno, A. (2008). Computer-based execution of clinical guidelines: A review. *International Journal of Medical Informatics*, *77*(12), 787–808.
11. Korb, K., & Nicholson, A. (2011). *Bayesian artificial intelligence* (2nd ed.). London: CRC Press.
12. Lam, H. P., Governatori, G., Satoh, K., & Hosobe, H. (2012). Distributed defeasible speculative reasoning in ambient environment. In M. Fisher, L. Torre, M. Dastani, & G. Governatori (Eds.), *Proceedings of the computational logic in multi-agent systems: 13th international workshop, CLIMA XIII*. Lecture notes in computer science (pp. 43–60). Berlin: Springer. doi:10.1007/978-3-642-32897-8_5.
13. Liu, Z., Malone, B., & Yuan, C. (2012). Empirical evaluation of scoring functions for Bayesian network model selection. *BMC Bioinformatics, 13*(Suppl 15), S14. doi:10.1186/1471-2105-13-S15-S14.
14. Oliveira, T., Neves, J., Novais, P., & Satoh, K. (2014). Applying speculative computation to guideline-based decision support systems. In *2014 IEEE 27th international symposium on computer-based medical systems (CBMS)* (pp. 42–47). IEEE. doi:10.1109/CBMS.2014.32.
15. Oliveira, T., Novais, P., & Neves, J. (2013). Representation of clinical practice guideline components in owl. In J. B. Pérez, J. M. C. Rodríguez, J. Fëhndrich, P. Mathieu, A. Campbell, M. C. Suarez-Figueroa, A. Ortega, E. Adam, E. Navarro, R. Hermoso, & M. N. Moreno (Eds.), *Trends in practical applications of agents and multiagent systems. Advances in intelligent systems and computing* (Vol. 221, pp. 77–85). Springer. doi:10.1007/978-3-319-00563-8_10.
16. Peleg, M. (2013). Computer-interpretable clinical guidelines: A methodological review. *Journal of Biomedical Informatics*, *46*(4), 744–763.
17. Prakken, H., & Sartor, G. (1997). Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, *7*(1–2), 25–75.
18. Russell, L., Gold, M., Siegel, J., Daniels, N., & Weinstein, M. (1996). The role of cost-effectiveness analysis in health and medicine. *JAMA*, *276*(14), 1172–1177. doi:10.1001/jama.1996.03540140060028.
19. Sadri, F., & Toni, F. (2006). Interleaving belief updating and reasoning in abductive logic programming. *Frontiers in Artificial Intelligence and Applications*, *141*, 442–446.
20. Saraswat, V., Jagadeesan, R., & Gupta, V. (1996). Timed default concurrent constraint programming. *Journal of Symbolic Computation*, *22*(5), 475–520.
21. Satoh, K. (2005). Speculative computation and abduction for an autonomous agent. *IEICE—Transactions on Information and Systems, E88-D*(9), 2031–2038.
22. Satoh, K., Codognet, P., & Hosobe, H. (2003). Speculative constraint processing in multi-agent systems. In *Intelligent agents and multi-agent systems* (Vol. 2891, pp. 133–144). Berlin: Springer.
23. Satoh, K., Inoue, K., Iwanuma, K., & Sakama, C. (2000). Speculative computation by abduction under incomplete communication environments. In *Proceedings of the fourth international conference on multi-agent systems* (Vol. 12, pp. 263–270). Alamitos: IEEE.
24. Satoh, K., & Yamamoto, K. (2002). Speculative computation with multi-agent belief revision. In *Proceedings of the first international joint conference on autonomous agents and multi-agent systems: Part 2, AAMAS '02* (pp. 897–904). New York: ACM.
25. Scutari, M. (2010). Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, *35*(3), 1–22.
26. Smolka, G. (1997). The oz programming model. In M. Broy & B. Schieder (Eds.), *Mathematical methods in program development*. NATO ASI series (Vol. 158, pp. 409–432). Berlin: Springer.
27. Ten Teije, A., Miksch, S., & Lucas, P. (2008). *Computer-based medical guidelines and protocols: A primer and current trends* (Vol. 139). Amsterdam: IOS Press.
28. Van De Wetering, G., Woertman, W. H., & Adang, E. M. (2012). Time to incorporate time in cost-effectiveness analysis. *European Journal of Health Economics*, *13*(3), 223–226. doi:10.1007/s10198-011-0374-3.
29. Van der Heijden, M., & Lucas, P. (2013). Describing disease processes using a probabilistic logic of qualitative time. *Artificial Intelligence in Medicine*, *59*(3), 143–155.
30. Visscher, S., Lucas, P., Schurink, C., & Bonten, M. (2009). Modelling treatment effects in a clinical Bayesian network using boolean threshold functions. *Artificial Intelligence in Medicine*, *46*(3), 251–266. doi:10.1016/j.artmed.2008.11.006.
31. Weber, P., Medina-Oliva, G., Simon, C., & Iung, B. (2012). Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. *Engineering Applications of Artificial Intelligence*, *25*(4), 671–682.
32. Witten, I., Frank, E., & Hall, M. (2011). *Data mining: Practical machine learning tools and techniques* (3rd ed.). San Francisco: Morgan Kaufmann.

2.6 A MOBILE AND EVOLVING TOOL TO PREDICT COLORECTAL CANCER SURVIVABILITY

Title	A Mobile and Evolving Tool to Predict Colorectal Cancer Survivability
Authors	Ana Silva, Tiago Oliveira, Vicente Julian, José Neves, and Paulo Novais
Publication Type	Conference proceedings
Chapter	Artificial Intelligence Applications and Innovations
Book Series	IFIP Advances in Information and Communication Technology
Conference	12th IFIP International Conference on Artificial Intelligence Applications and Innovations
Publisher	Springer International Publishing Switzerland
Volume	475
Pages	14-26
Year	2016
Online ISBN	978-3-319-44944-9
Print ISBN	978-3-319-44943-2
URL	http://link.springer.com/chapter/10.1007/978-3-319-44944-9_2
State	Published
Scimago journal rank (2015 - last known)	0.160, Information Systems and Management (Q4)
Indexation	Scopus, DBLP

Contribution of the doctoral candidate
The doctoral candidate, Tiago José Martins Oliveira, declares to be one of the main authors and contributors of the paper *A Mobile and Evolving Tool to Predict Colorectal Cancer Survivability*.

A Mobile and Evolving Tool to Predict Colorectal Cancer Survivability

Ana Silva[1], Tiago Oliveira[1](✉), Vicente Julian[2], José Neves[1],
and Paulo Novais[1]

[1] Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
a55865@alunos.uminho.pt, {toliveira,jneves,pjon}@di.uminho.pt
[2] Departamento de Sistemas Informáticos y Computación,
Universidad Politécnica de Valencia, Valencia, Spain
vinglada@dsic.upv.es

Abstract. In this work, a tool for the survivability prediction of patients with colon or rectal cancer, up to five years after diagnosis and treatment, is presented. Indeed, an accurate survivability prediction is a difficult task for health care professionals and of high concern to patients, so that they can make the most of the rest of their lives. The distinguishing features of the tool include a balance between the number of necessary inputs and prediction performance, being mobile-friendly, and featuring an online learning component that enables the automatic evolution of the prediction models upon the addition of new cases.

1 Introduction

The colorectal cancer, is a subtype of cancer, which affects the lower portion of the gastrointestinal tract and develops in the cells lining the colon and rectum [24]. It can be further divided according to the site where the pathology develops. Colon and rectum cancers are, in fact, different pathologies, with different associated genetic causes and different progressions according to distinct molecular pathways [29]. Statistics show that colorectal cancer is the most common form of cancer in the digestive system, the third most common and the fourth deadliest cancer overall [15].

The use of machine learning (ML) techniques has been growing in cancer research [17]. The accurate prediction of survivability in patients with cancer remains a challenge namely due to the heterogeneity and complexity of the disease. However, accurate survivability prediction is important for patients with cancer so that they can make the most of the rest of their lives. It is also important to help clinicians to make the best decisions, when palliative care is an essential component of the process. Given that colon and rectal are the most common cancers of the digestive system, one would expect the existence of numerous tools for ascertaining the likelihood of a patient surviving this disease. Although there are some tools for this task, few provide predictions for both colon and rectal cancer, and none of them apply ML techniques in order to

L. Iliadis and I. Maglogiannis (Eds.): AIAI 2016, IFIP AICT 475, pp. 14–26, 2016.
DOI: 10.1007/978-3-319-44944-9_2

build evolving predictive models. Furthermore, their digital support may hinder their consultation at care delivery.

The objective of this work is to present an easy to use tool that provides survivability predictions of colon and rectal cancer patients for 1, 2, 3, 4 and 5 years after diagnosis and treatment. Due to the ubiquitous presence of mobile devices in everyday life and the ease with which one is able to consult these devices and use their applications, we chose to develop this tool as a mobile application. The underlying model for survivability prediction was obtained through ML techniques applied to the data from the Surveillance, Epidemiology, and End Results (SEER) program [18], a large cancer registry in the United States, and arguably the most complete cancer database in the world. The dataset includes records of patients diagnosed with different types of cancer from 1973 to 2012. The focus of this paper will be placed on the mobile solution developed for survivability prediction, but a part of the paper will be dedicated to briefly describing its underlying ML model so as to provide a better comprehension of the work as a whole.

The paper is structured as follows. Section 2 presents related work featuring survivability prediction tools for colon and rectal cancer, with an analysis of their main strengths and limitations. Section 3 describes the selected requirements for the tool, its underlying ML-based predictive model, architecture and a comprehensive use case. Section 4 provides a reflection about the strengths and limitations of our approach. Finally, Sect. 5 presents the conclusions drawn so far and future work considerations.

2 Related Work

Existing tools for colon or rectal cancer survivability prediction are mostly available as web applications. Table 1 shows a summary of their main features, namely: (1) whether the application is used for colon or rectal cancer; (2) the number of features necessary to get a prediction; (3) the data set that its underlying model is based on; (4) the technique used to construct the predictive model; (5) the type of target prediction it produces; and (6) a measure of performance in the form of a concordance index (C-index). The C-index corresponds to the probability of giving a correct response in a binary prediction problem. It is considered to be numerically equivalent to the area under the ROC curve (AUC) [16].

There is a disparity in the number of features used in each tool. However, twelve [22] or even nine [10,25] features may be too much information for a physician to input on-the-fly. Furthermore, there are cases in which the increased number of features does not necessarily translate into a better performance, as can be seen in the direct comparison between the works in [28] and in [22].

All the underlying models are based on statistical modelling, most notably on Cox regression analysis [8]. This is the dominant multivariate approach used in survivability prediction and corresponds to a multiple linear regression of the hazard on a set of variables. This indicates that the use of soft computing techniques, namely ML, in survivability prediction, especially in colon and rectal

Table 1. Characteristics of applications for colon and rectal cancer survivability prediction.

Characteristics	Bush and Michaelson [10]	Chang et al. [11]	Weiser et al. [28]	Renfro et al. [22]	Wang et al. [27]	Valentini et al. [25]	Bowles et al. [7]
(1) Cancer type	Colon	Colon	Colon	Colon	Rectal	Rectal	Rectal
(2) Number of Features	9	6[a]	2/3/7	12	5[a]	9	7[a]
(3) Data set	SEER	SEER	SEER	Adjuvant Colon Cancer End Points (ACCENT)	SEER	Five European randomized trials	SEER
(4) Model	Regression based	Regression based	Regression based	Regression based	Regression based	Regression based	Regression based
(5) Target	0 – 15 years	1 – 10 years (disease specific survivability) 0 – 5 years (conditional survivability)	5 years	5 years	0 – 5 years	1–10 years	1 – 10 years (disease specific survivability) 0 – 5 years (conditional survivability)
(6) Performance C-index	–	0.816	0.61/0.63/0.68	0.66	0.75	0.70	–

[a] Including months which the patient has already survived (for conditional survivability calculation).

cancer, has yet to be fully explored. Since one of the advantages of ML is having more discriminative power in identifying patterns in data and finding nuances that may escape statistical modelling, its usage for survivability prediction may result in models with better performances [17]. As such, ML was chosen as the modelling approach for this work.

Most of the target predictions, either for colon or for rectal cancer, cover a 5-year span [22,27,28]. Even though there are models that cover a wider time span [7,10,11,25], the five year barrier is an important goal for a colorectal cancer patient to overcome, and is used throughout clinical practice guidelines [4,5] as a turning point for follow-up procedures, in which the vigilance over the patient is lightened, and for the assessment of the recurrence risk. For this reason, the present work will also have a target prediction of five years. Another noteworthy observation is that only two of the tools feature conditional survivability predictions.

To determine if the tools are suitable for mobile devices, the applications were analysed using the mobile-friendly test tool from *Google*[1]. The results showed that, except for the tools reported in [22,28], all the others are unsuitable for mobile access. The test revealed that the text was too small to read, the mobile viewport was not set, links were too close to each other and usually the content

[1] Mobile-friendly test tool of Google is available at https://www.google.com/webmasters/tools/mobilefriendly/.

was wider than the screen. Therefore, few of these applications had a mobile-friendly design. Another goal is to address this by developing a cross-platform tool that is available to users in a practical and intuitive way, through a smartphone or tablet.

3 CRCPredictor: An Application for Survivability Prediction

Throughout the last decade, mobile phones have gone from being simple phones to being handheld pocket-sized computers. Their capabilities, namely the processing and on-board computing capacity incite the development of applications [6]. According to data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, the Android of Google and iOS of Apple are the two most popular smartphone operating systems [12].

For the health care industry, mobile applications yielded new boundaries in providing better care and services to patients. Moreover, it is making a revolution in the way information is managed and made available [23]. The portability of mobile applications can increase the productivity of health care professionals. It grants a rapid access to information and multimedia resources, allowing health care professionals to make decisions more quickly with a lower error rate, increasing the quality of patient documentation and improved workflow patterns [26]. This work discloses an assistive tool to help physicians to improve their practice. The problem it addresses is predicting the survivability of colorectal cancer patients in an individualized manner.

3.1 Requirements for the Survivability Prediction Tool

Several functionalities were delineated to achieve a solution that covers the limitations mentioned in Sect. 2 and, at the same time, is able to help physicians to improve their practice. These functionalities are summarized in the following functional requirements for the prediction tool: allow the user to select the cancer type (either colon or rectal) for which he seeks a prediction; allow the user to provide inputs for a set of selected features, based on which the underlying models generate survivability predictions; allow the user to choose the value of an input for a feature from a set of pre-determined values; provide a survivability prediction, according to the inputs, for 1, 2, 3, 4 and 5 years after the diagnosis and treatment; provide a likelihood value for the prediction of each year; to allow the visualization of the predictions and likelihood values in a chart; and allow the insertion of new patient registries into the case database, thus increasing the number of cases for the periodic recalculation of the prediction models.

Additional requirements for the tool are that it should be made available in the two main mobile platforms (iOS and Android) and it should be able to recalculate the prediction models upon the addition of a significant number of new patient registries. This confers a dynamism to the prediction models and should ensure their evolution over time.

3.2 Colon and Rectal Cancer Survivability Prediction Models

Survivability prediction was approached as a binary classification problem. The goal was to produce predictions for 1, 2, 3, 4 and 5 years after treatment of colon or rectal cancer. Each classification label (there were five representing years 1, 2, 3, 4 and 5) could only have two values: *survived* or *not survived*. As such, it was necessary to build five survivability prediction models (one per year) for each type of cancer. The created models were based on the SEER dataset. The criteria for selecting patient registries was the same for both colon and rectal cancer. Only patients with age greater than or equal to 18 years old were selected. Patients who were alive at the end of the data collection whose survival time had not yet reached 60 months (five years) and those who passed away of causes other than colon or rectal cancer were sampled out. After preprocessing, 38,592 cases were isolated for colon cancer and 12,818 cases were considered for rectal cancer. From the isolated cases for each pathology, 10 % were selected for testing sets. After filtering cases with "unknown" values, the colon cancer testing set had 2,221 cases and the training set had 20,061 cases. The testing set for rectal cancer had 551 cases and the training set had 4,962 cases. In total, the training set had 61 attributes representing possible classification features.

All the phases, from preprocessing to evaluation, were executed using RapidMiner[2], an open source data mining software chosen for its workflow-based interface and an intuitive application programming interface (API).

Using the Optimize Selection [21] operator for feature selection with the classification labels as target, a total of 6 features were obtained from a feature selection phase for each cancer type. Their name and description are shown in Tables 2 and 3. The training sets for colon and rectal cancer with their respective selected features were used in the learning of multiple prediction models using different ML ensemble methods such as bagging, adaboost, bayesian boosting, stacking, and voting. The accuracy, the AUC and the F-measure were used as performance measures in order to evaluate the models developed for colon and rectal cancer. The accuracy is the percentage of correct responses among the examined cases [9]. The F-measure is a combination of precision (a form of accuracy, also known as positive predictive value) and recall (also known as sensitivity) [20]. The AUC can be interpreted as the percentage of randomly drawn data pairs of individuals that have been accurately classified in the two populations [16]. These measures were calculated using the training data set and 10-fold cross validation. By applying the testing sets to the models, we calculated the percentage of incorrectly classified cases. The stacking[3] [14], using k-NN, decision tree, and random forest classifiers as base learners and a naive bayes classifier as a stacking model learner, was the best performing model for both colon and rectal cancer. Upon prediction, the model is capable of providing a

[2] Software available at https://rapidminer.com.

[3] Stacking combines base classifiers of different types. Each base classifier generates a model using the training set, then a meta-learner integrates the independently learned base classifier models into a high level classifier by re-learning a meta-level training set.

Table 2. Features obtained by feature selection and used for colon cancer models.

Attribute	Description
Age at diagnosis	The age (in years) of the patient at time of diagnosis
Carcinoembryonic antigen	The interpretation of the highest Carcinoembryonic Antigen test results
Clinical assessment of regional lymph nodes	The clinically evident regional lymph nodes
AJCC stage	The grouping of the TNM information combined
Primary site	Identification of the site in which the primary tumor originated
Regional nodes examined	The total number of regional lymph nodes that were removed and examined by the pathologist

Table 3. Features obtained by feature selection and used for rectal cancer models.

Attribute	Description
Age at diagnosis	*
Extension of the Tumor	Information on extension of the tumor
Tumor size	Information on tumor size (in mm)
AJCC stage	*
Surgery of primary site	Describes a surgical procedure that removes and/or destroys tissue of the primary site performed as part of the initial work-up or first course of therapy.
Gender	The sex/gender of the patient at diagnosis

*Described in Table 2.

confidence value that represents the likelihood of the prediction. Table 4 shows the performance values of the best model developed, for both cancer types.

As the intent with this paper is to present the features of the developed tool and describe its inner workings, it was considered that an exhaustive description of the ML process was out of scope.

3.3 Architecture

The CRCPredictor is a hybrid mobile application targeting smartphones and tablets. The back-end of this tool includes two web services: one to give the survivability prediction responses for colon or rectal cancer to the user and another to recalculate the survivability prediction models. Figure 1 shows the architecture of the CRCPredictor system.

The *Survival Prediction App* was developed using a hybrid approach, between a web and a native methodology. This allows an abstraction from the native language of the target operating system while retaining the core features of a native app. A hybrid application is developed by applying web technologies

Table 4. Performance measures for 10-fold cross validation and the incorrectly classified cases from the test data set of the stacking model.

Cancer type	Measure	1 Year	2 Year	3 Year	4 Year	5 Year	Average
Colon	AUC	0.982	0.985	0.987	0.989	0.987	0.986
	Accuracy	95.84 %	96.62 %	96.79 %	97.09 %	97.03 %	96.67 %
	F-Measure	90.41 %	94.43 %	95.55 %	97.61 %	97.51 %	95.10 %
	Incorrectly classified (%)	3.24 %	3.02 %	2.97 %	2.84 %	3.11 %	3.03 %
Rectal	AUC	0.961	0.973	0.973	0.975	0.969	0.970
	Accuracy	95.61 %	95.81 %	93.81 %	94.82 %	94.32 %	94,87 %
	F-Measure	80.56 %	87.32 %	85.85 %	96.63 %	96.20 %	89.31 %
	Incorrectly classified (%)	3.81 %	3.99 %	6.17 %	4.36 %	4.36 %	4.54 %

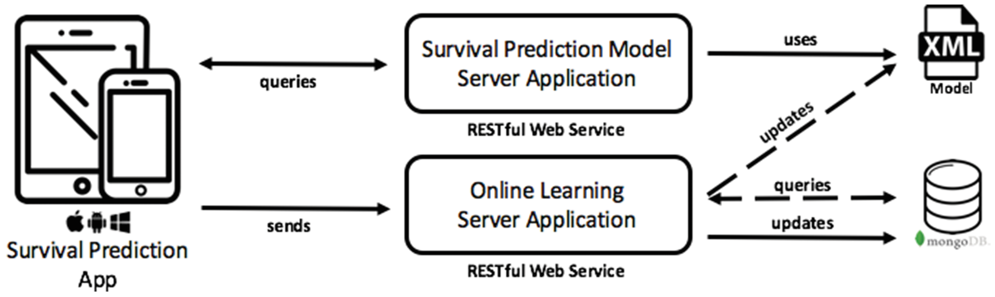


Fig. 1. Architecture of the CRCPredictor system.

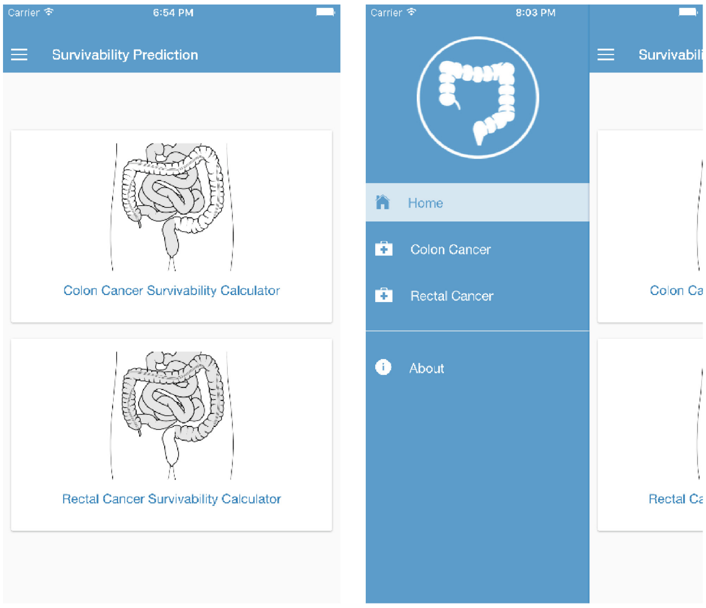
(mainly, HTML5, CSS and JavaScript) and is executed inside a native container on the mobile device. It is suitable for multiple platforms and is distributable through an application store, just like native applications. This type of approach can have an inferior performance compared with native applications. However, nowadays mobile devices have powerful capabilities and the performance gap is hardly noted. The application was developed using AngularJS, Ionic Framework, and Cordova. Cordova wraps the HTML/JavaScript app into a native container which can access the device functions of several platforms [1]. These functions are exposed via a unified JavaScript API, for an easy access to the full native functionalities.

The *Survival Prediction Model Server Application* was developed to cover the need of an individualized system, able to respond according to a particular set of patient characteristics. It exposes a set of RESTful web services. This service architecture was chosen for being light-weight, easy to access and scalable [30]. The web services were developed in Java with the Java API for RESTful Web Services (JAX-RS) [2]. The data is sent over the HTML POST method when the health care professional submits the values for the prediction features on the *Survival Prediction App*. The RESTful web service, using the RapidMiner API, receives the values and feeds them to the corresponding models, encoded in XML files. The response with the survivability predictions for the five years is returned in a JSON format.

The *Online Learning Server Application* also follows a REST architecture. It handles newly submitted patient data. The outcomes are added to a database for a posterior recalculation of all the models, which keeps them up-to-date. The data is inserted into a NoSQL database and, for each 1000 new registries, the models for the five years are recalculated, generating five new XML files for the type of cancer that just got the thousandth new case. The 1000 mark was arbitrarily defined and can be subject to adjustment.

3.4 Use Case

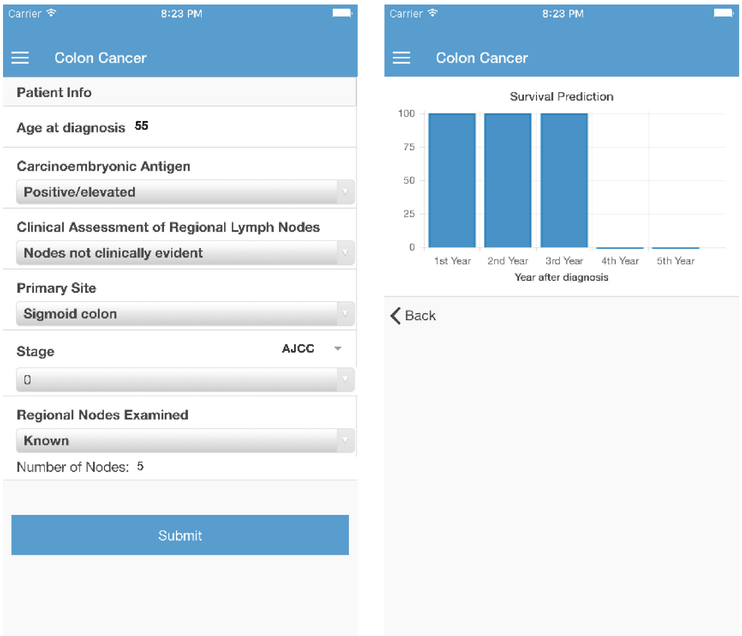
Figure 2a shows the first screen that appears when the *Survival Prediction App* of the CRCPredictor is initiated. By clicking on the menu (Fig. 2b), all options available in this application become visible.



(a) Home screen. (b) Menu.

Fig. 2. Home screen and menu of the Survival Prediction App of the CRCPredictor.

A typical use case is getting a prediction for colon cancer survivability. Supposing a physician is treating a patient diagnosed with colon cancer, once the type of cancer in the home screen is set (as shown in Fig. 2b), the health care professional inserts the values for the selected features (Fig. 3a). All features, except for the age of the patient, are filled in by choosing the value from a list of available options. By submitting a case of a patient with *55* years old, having a *positive/elevated* carcinoembryonic antigen value, with clinical assessment of regional lymph nodes of *not clinically evident*, with the primary site of the cancer being in the *sigmoid colon*, with stage *0* and with *5* as the number of regional nodes examined, the values are sent to the *Survival Prediction Model Server Application* and the outcome is calculated. The prediction is always provided in the form of confidence values for a positive prediction, i.e., the confidence that the patient will survive. This is displayed in a new screen in the form of a bar chart (Fig. 3b). For the stage of the patient, the physician can choose between the TNM system or the grouped stage, known as American Joint Committee on Cancer (AJCC) stage. The results show that, while the model was able to predict with 100 % confidence that the patient will survive the first three years, the confidence of his surviving the fourth and fifth years is 0 %. To predict the survivability of a patient diagnosed with rectal cancer, the procedure is similar to the one used for colon cancer.



Colon Cancer

Patient Info

Age at diagnosis 55

Carcinoembryonic Antigen: Positive/elevated

Clinical Assessment of Regional Lymph Nodes: Nodes not clinically evident

Primary Site: Sigmoid colon

Stage: AJCC 0

Regional Nodes Examined: Known

Number of Nodes: 5

(a) Survivability Features of Colon Cancer.

(b) Results for Colon Cancer.

Fig. 3. Colon cancer survivability calculator.

4 Analysis and Discussion

In terms of inputs the constructed prediction models, for both colon and rectal cancer, require only the input of six selected features. Comparing with the related tools in Sect. 2, this number is inferior to the number of features used in the underlying models of the tools described in [7,10,22,25], and is closer to the number of inputs of the remaining prediction tools. The number of input features may be crucial to the adoption or the rejection of a tool, as it may become difficult to use it on-the-fly if too much information is needed. Another aspect to note is that, apart from the *age at diagnosis* and the *AJCC Stage*, the feature selection produced two very different sets for colon and for rectal cancer, which is in line with the notion that the two, although having aspects in common, are different pathologies. Regarding the colon cancer features, the *age at diagnosis* and the *AJCC stage* are present in most colon cancer prediction tools [11,22,28]. The other selected features are not usually present, but they are closely related to the ones that are. For instance, the *clinical assessment of regional lymph nodes* is a product of a medical evaluation of a feature widely used in the existing tools [22,28] that is the number of lymph nodes found to have cancer out of the lymph nodes isolated during surgery. The same can be said about the selected features for rectal cancer, i.e., they are, at the very least, closely related to the ones used in other prediction tools.

In [3], the use of ML ensemble models to develop survival prediction models for colon cancer is described. The modelling component of our work is similar to that approach; therefore, it is possible to compare the performance of our selected model with theirs. The classification accuracies reported in [3] for years 1, 2, and 5 were 90.38 %, 88.01 %, and 85.13 %. The reported AUCs were 0.96, 0.95, and 0.92, respectively. As shown in Table 4, we were able to improve the classification results with our models. This direct comparison is not possible for the rectal cancer model as it was not possible to find such a closely related work in the literature. However, it is possible to verify that the rectal cancer model performs worse than the colon model in every metric, possibly due to the smaller size of the training set used in the learning process. At the same time, both the colon and rectal cancer models showed low classification errors on the randomly selected test data sets. Additionally, when comparing the AUCs of the generated models in Table 4 with the C-indexes in Table 1, it is possible to conclude that the generated models show a better discriminative power than the currently available models.

Regarding the CRCPredictor system, it fulfils the requirements defined at the beginning of the work. The distinguishing features of the system's architecture are its flexibility and scalability, which make the addition of new features (services) simple and easy. The *Survival Prediction App* was developed as a mobile-friendly application, enabling the easy access of health care professionals to its functionalities on their mobile devices. Another component that distinguishes this system from established tools is the *Online Learning Server Application* which ensures the continuous evolution of the prediction models. However, the system does not provide conditional survivability predictions, which makes it less appealing when compared with the works in [7,11,25], as this is a type of information that health care professionals generally like to know.

5 Conclusions and Future Work

The main contribution of this work is a survivability prediction tool for colon and rectal cancer. Its distinguishing features are a balance between the number of necessary inputs and prediction performance, being mobile-friendly, and featuring an online learning component that enables the automatic recalculation and evolution of the prediction models upon the addition of new cases. The goal with this tool is to facilitate the access of health care professionals to instruments capable of enriching their practice and improving their results. Future work on the tool includes the development of conditional survivability models that allow the user to get a prediction knowing that the patient has already survived a number of years after diagnosis and treatment. Additionally, we intend to conduct experiments to assess how well the tool fulfils the needs of health care professionals and identify aspects to improve. Additionally, the models presented herein will be considered for inclusion in a guideline-based decision support system, described in [19], as a dynamic knowledge complement to the static recommendations of clinical practice guidelines. Since colon and rectal cancer affect mostly the elderly, this survivbility prediction apllication can be used within a technological environment, such as the one disclosed in [13], to provide better support to this population group.

Acknowledgements. This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope UID/CEC/00319/2013. The work of Tiago Oliveira is supported by a FCT grant with the reference SFRH/BD/85291/2012.

References

1. Apache cordova. http://cordova.apache.org/. Accessed 31 Mar 2016
2. Project kenai. https://jax-rs-spec.java.net/. Accessed 04 Apr 2016
3. Al-Bahrani, R., Agrawal, A., Choudhary, A.: Colon cancer survival prediction using ensemble data mining on seer data. In: 2013 IEEE International Conference on Big Data, pp. 9–16 (2013)
4. Benson, A., Bekaii-Saab, T., Chan, E., Chen, Y.J., Choti, M., Cooper, H., Engstrom, P.: NCCN clinical practice guideline in oncology colon cancer. Technical report, National Comprehensive Cancer Network (2013)
5. Benson, A., Bekaii-Saab, T., Chan, E., Chen, Y.J., Choti, M., Cooper, H., Engstrom, P.: NCCN clinical practice guideline in oncology rectal cancer. Technical report, National Comprehensive Cancer Network (2013)
6. Boulos, M.N., Wheeler, S., Tavares, C., Jones, R.: How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from ecaalyx. Biomed. Eng. Online **10**(1), 24 (2011)
7. Bowles, T.L., Hu, C.Y., You, N.Y., Skibber, J.M., Rodriguez-Bigas, M.A., Chang, G.J.: An individualized conditional survival calculator for patients with rectal cancer. Dis. Colon Rectum **56**(5), 551–559 (2013)
8. Bradburn, M.J., Clark, T.G., Love, S.B., Altman, D.G.: Survival analysis part II: multivariate data analysis - an introduction to concepts and methods. Br. J. Cancer **89**(3), 431–436 (2003)

9. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn. **30**(7), 1145–1159 (1997)
10. Bush, D.M., Michaelson, J.S.: Derivation: Nodes + PrognosticFactors Equation for Colon Cancer accuracy of the Nodes + PrognosticFactors equation. Technical report (2009)
11. Chang, G.J., Hu, C.Y., Eng, C., et al.: Practical application of a calculator for conditional survival in colon cancer. J. Clin. Oncol. **27**(35), 5938–5943 (2009)
12. Corporation, I.D.: Idc: Smartphone os market share 2015, 2014, 2013, and 2012. http://www.idc.com/prodserv/smartphone-os-market-share.jsp. Accessed 30 Mar 2016
13. Costa, R., Novais, P., Machado, J., Alberto, C., Neves, J.: Inter-organization cooperation for care of the elderly. In: Wang, W., Li, Y., Duan, Z., Yan, L., Li, H., Yang, X. (eds.) Integration and Innovation Orient to E-Society Volume 2. IFIP, vol. 252, pp. 200–208. Springer, Boston (2007)
14. Džeroski, S., Ženko, B.: Is combining classifiers with stacking better than selecting the best one? Mach. Learn. **54**(3), 255–273 (2004)
15. Ferlay, J., Soerjomataram, I., Ervik, M., et al.: Globocan 2012: Estimated cancer incidence, mortality and prevalence worldwide in 2012 (2012). http://globocan.iarc.fr. Accessed 27 Dec 2015
16. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology **143**(1), 29–36 (1982)
17. Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V., Fotiadis, D.I.: Machine learning applications in cancer prognosis and prediction. Comput. Struct. Biotechnol. J. **13**, 8–17 (2015)
18. National Cancer Institute: Surveillance, epidemiology and end results program (2015). http://seer.cancer.gov/data/. Accessed 10 Jan 2016
19. Novais, P., Oliveira, T., Neves, J.: Moving towards a new paradigm of creation, dissemination, and application of computer-interpretable medical knowledge. Prog. Artif. Intell. **5**(2), 77–83 (2016)
20. Powers, D.M.W.: What the F-measure doesn't measure .... Technical report, Beijing University of Technology, China & Flinders University, Australia
21. RapidMiner: Rapidminer documentation: Optimize selection (2016). http://docs.rapidminer.com/studio/operators/data_transformation/attribute_space_transformation/selection/optimization/optimize_selection.html. Accessed 03 Jan 2016
22. Renfro, L.A., Grothey, A., Xue, Y., Saltz, L.B., André, T., Twelves, C., Labianca, R., Allegra, C.J., Alberts, S.R., Loprinzi, C.L., et al.: Accent-based web calculators to predict recurrence and overall survival in stage III colon cancer. J. Natl. Cancer Inst. **106**(12), dju333 (2014)
23. Siau, K., Shen, Z.: Mobile healthcare informatics. Med. Inform. Internet Med. **31**(2), 89–99 (2006)
24. Vachani, C., Prechtel-Dunphy, E.: All about rectal cancer (2015). http://www.oncolink.org/types/article.cfm?aid=108&id=9457&c=703. Accessed 27 Dec 2015
25. Valentini, V., van Stiphout, R.G., Lammering, G., Gambacorta, M.A., Barba, M.C., Bebenek, M., Bonnetain, F., Bosset, J.F., Bujko, K., Cionini, L., Gerard, J.P., Rödel, C., Sainato, A., Sauer, R., Minsky, B.D., Collette, L., Lambin, P.: Nomograms for predicting local recurrence, distant metastases, and overall survival for patients with locally advanced rectal cancer on the basis of European randomized clinical trials. J. Clin. Oncol. **29**(23), 3163–3172 (2011)
26. Ventola, C.L.: Mobile devices and apps for health care professionals: uses and benefits. Pharm. Ther. **39**(5), 356–364 (2014)

27. Wang, S.J., Wissel, A.R., Luh, J.Y., Fuller, C.D., Kalpathy-Cramer, J., Thomas, C.R.: An interactive tool for individualized estimation of conditional survival in rectal cancer. Ann. Surg. Oncol. **18**(6), 1547–1552 (2011)
28. Weiser, M.R., Gönen, M., Chou, J.F., Kattan, M.W., Schrag, D.: Predicting survival after curative colectomy for cancer: individualizing colon cancer staging. J. Clin. Oncol. **29**(36), 4796–4802 (2011)
29. Yamauchi, M., Lochhead, P., Morikawa, T., et al.: Colorectal cancer: a tale of two sides or a continuum? Gut **61**(6), 794–797 (2012)
30. Zhao, H., Doshi, P.: Towards automated restful web service composition. In: IEEE International Conference on Web Services, ICWS 2009, pp. 189–196. IEEE (2009)

3

CONCLUSIONS

Throughout the work for this doctoral thesis, the doctoral candidate aimed to tackle pressing matters in the development of CIG-based CDSSs. While it is true that it was only possible to approach some aspects of these systems, it can be considered that the CompGuide framework, in the form of its many solutions, such as the ontological model for CIGs, the architecture for CIG execution, the health care assistant, the guideline editor (mentioned further below), the mechanism for managing incomplete information in decision-making based on Speculative Computation, and the survivability prediction model, address the limitations that were identified in the beginning of the document.

This chapter provides an analysis of the contributions of this doctoral thesis in light of the objectives outlined initially. The content of this chapter then proceeds to explain how the research hypothesis formulated at the beginning of the thesis was validated by the obtained results. There is also a summary of the activities that were undertaken for the dissemination of results. The chapter ends with final remarks regarding the work that was developed so far and with future work considerations.

3.1 CONTRIBUTIONS

The completion of this doctoral thesis in the form of the publications comprised in Chapter 2, resulted in important contributions. Their sum makes the CompGuide framework. As a complement to this section, Table 1 shows the objectives for the thesis, as they were defined in Section 1.5, and the sections of Chapter 2 where they were achieved.

The full list of contributions is the following:

- **Identification of relevant issues and respective solutions in CIG development.**

The first contribution, which is in Section 2.1, was the characterization of the landscape of CIG development and the identification of relevant issues to be addressed. Here, it was possible to observe that AI techniques can provide solutions for these problems, which include the lack of comprehensive approaches to CIGs that are equally focused in aspects such as the definition of clinical recommendations, clinical constraints, temporal constraints, and control sequences within CPGs. The complexity of current CIG model approaches and the lack of methods to smooth this complexity in order to facilitate the acquisition of knowledge were also identified as limiting factors for the adoption of CIG-based CDSSs. The decision support features in CIG execution systems were also observed to be limited in the sense that they did not provide mechanisms to deal with uncertainty for data entry points during the execution.

The main conclusion from this section is that CIGs are still far from the concept of *living guidelines*, as they do not show the flexibility and adaptability that characterize dynamic, modular, and interactive approaches. AI techniques were identified as solutions to explore in order to address these issues. The formalization of CPGs in an ontology, using Web Ontology Language (OWL) in particular, was identified as addressing the existing problems in GIG representation, namely the lack of flexibility. Furthermore, the use of web technologies to develop more responsive and interactive systems was defined as the path to follow in order to increase the intervention of CIGs in a clinical setting. As for adaptability, it was proposed that the combination of CBR with logic programming integrated in the decision-making process of CIGs would be an appropriate way to handle data-centred uncertainty. While CBR was abandoned later on in the work, the idea that previous CIG executions and the data retrieved from them could benefit future executions was kept.

Given the elements included in this contribution, it can be considered that Objective 1 and Objective 2 (see Table 1) were fully accomplished.

- **A comprehensive model for CIGs covering the different content of CPGs.**

This contribution is provided in Section 2.2 and it consists of a set of knowledge elements formalized in an OWL ontology. This ontology for CPG representation was called CompGuide and follows a Task Network Model (TNM) in which all recommendations are modelled as *Plans*, *Actions*, *Questions* and *Decisions*. The model covers the representation of administrative information, the definition of workflows of procedures, the definition of temporal constraints, and the definition of clinical constraints. In this sense, the CompGuide ontology is shown to be encompassing, and presents advantages when compared to CIG approaches such as Arden Syntax, GLIF3, or PROforma. A distinguishing feature of the model is that it does not require proficiency in any programming language, and thus, reduces the the requirements a user must have for the acquisition of CIGs. Everything is modelled using the basic components of an ontology, namely classes, properties, and individuals. The development of the ontology proceeded and was divulged in subsequent publications. The most notorious update was the improvement of the temporal model in (Oliveira et al., 2017). The end result is an ontology with 33 classes, 37 object properties, and 54 data properties.

The CIG ontology resulting from this contribution is one of the elements that allowed the fulfilment of Objective 3 (see Table 1).

- **A CIG model whose expressiveness covers a wide range of CPGs.**

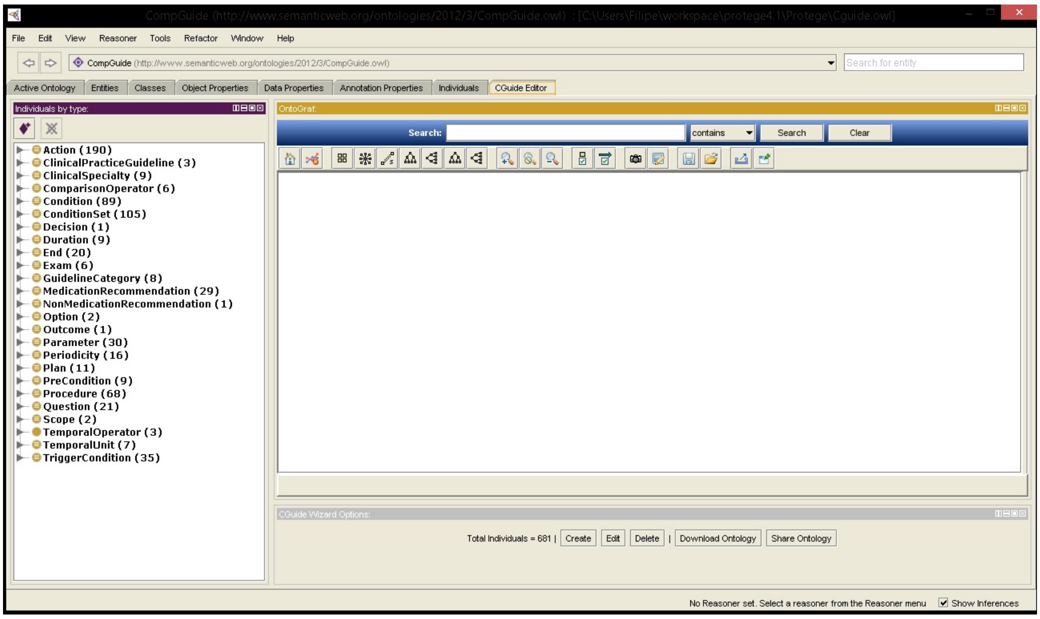
According to the experiment described in Section 2.3. When assessed with a set of different CPGs, from different clinical specialities and with different categories, it was considered that the CompGuide ontology allows the representation of series of questions, decisions and actions in a way that mimics the organization of the algorithms of guidelines. Furthermore, there was a wide agreement that the CompGuide ontology could effectively be used to represent the CPGs in question. There were also suggestions to improve the ontology, mainly concerning the representation of medication prescriptions, values in the specification of clinical conditions, and temporal restrictions. This criticism was assimilated and these aspects were further improved in the ontology.

Given the overall appreciation of the CompGuide ontology, it can be considered that the model covers a wide range of CPGs, and thus, Objective 3 (see Table 1) was accomplished.

- **A method for CIG acquisition specifically tailored for an ontology.**

The process of encoding a CPG in the CompGuide ontology, as it is described in Section 2.2, automatically generates knowledge patterns that can be reused in the definition of other parts of the same CIG. However it is necessary to tailor this acquisition to take full advantage of this fact. This contribution corresponds to the implementation of a step-by-step method that allows the acquisition of CIGs from CPGs in the form of a tool based on Protégé 4. It was developed as a plugin (main window shown in Figure 4) that features a set of assistants for the creation and editing of CIGs. It reveals a set of advantageous characteristics such as automation in the management of knowledge elements in the ontology, facilities that allow to easily reuse knowledge already introduced in the representation, and the possibility to share CIGs in a cloud repository. Since, at the time of writing this thesis, there were no publications featuring this contribution, it was decided to describe the contribution herein.

Taking into account the features integrated in the CompGuide editor, it is considered that Objective 4 (see Table 1) was achieved.


CompGuide (http://www.semanticweb.org/ontologies/2012/3/CompGuide.owl) : [C:\Users\Filipe\workspace\protege4.1\Protege\Cguide.owl]

CompGuide (http://www.semanticweb.org/ontologies/2012/3/CompGuide.owl)

Individuals by type:

- Action (190)
- ClinicalPracticeGuideline (3)
- ClinicalSpecialty (9)
- ComparisonOperator (6)
- Condition (89)
- ConditionSet (105)
- Decision (1)
- Duration (9)
- End (20)
- Exam (6)
- GuidelineCategory (8)
- MedicationRecommendation (29)
- NonMedicationRecommendation (1)
- Option (2)
- Outcome (1)
- Parameter (30)
- Periodicity (16)
- Plan (11)
- PreCondition (9)
- Procedure (68)
- Question (21)
- Scope (2)
- TemporalOperator (3)
- TemporalUnit (7)
- TriggerCondition (35)

OntoGraf:

Search: contains

CGuide Wizard Options:

Total Individuals = 681 |

No Reasoner set. Select a reasoner from the Reasoner menu

☑ Show Inferences

Figure 4: Main window of the CompGuide editor for the acquisition of CIGs.

- **An architecture for the execution of CIGs, deployable in standalone applications.**

In Section 2.4 the service-oriented architecture of the CompGuide system is presented. It provides an API to access CIGs in the CompGuide ontology and the functionalities of the guideline execution engine. Other developers may use it in their own web or mobile applications. This way of making advice available offers the possibility to easily develop other user interfaces. It allows remote guideline execution with data centralization.

The idea of CPGs as a service, which is further explored in (Oliveira et al., 2014a) and (Novais et al., 2016), facilitates the deployment of CPGs, making them more accessible and their implementation more agile. Such an architecture fulfills Objective 5 (see Table 1).

- **A novel way to integrate CIG recommendations in the daily practice of health care professionals.**

The contribution resulted from the work presented in Section 2.4. The prototype application follows the principle of embedding the recommendations of a CIG-based CDSS in the activities of a health care professional. This principle is materialized in the form of a health care assistant. As such, besides all the functionalities of data input and reasoning according to the CPG logics, the tasks suggested by the system are mapped onto an agenda for the health care professional. This allows him to accurately track his activities while benefiting from automatic reasoning features, according to the clinical constraints defined in a CIG. A tool implementing this form of providing advice shows two behaviours, as described in Section 1.3, a tool for patient-specific recommendations and a tool for focusing attention.

As such, the way to achieve a higher integration of the advice from a CIG-based CDSS into the life of a health care professional is, from this point of view, to capture his attention and to assist him in keeping track of his tasks. This contribution accomplishes Objective 6 (see Table 1).

- **A method for the management of incomplete information in CIG decision-making steps.**

The method is based on Speculative Computation and is presented in Section 2.5. It relies on logic programming and default reasoning, consisting of three phases: process reduction phase, fact arrival phase and default revision phase. The method also integrates a dynamic revision of default constraints provided by Bayesian networks. Speculative Computation with Default Revision manages the state of the information, fulfilling the role of an interface to rule-guided problems, which is basically how the knowledge base of CIG systems is organized. This also results in clinical scenarios with the most likely outcomes of decisions when there is no information or there is partial information. Due to this mechanism, CIG execution may carry on, even in the presence of this form of uncertainty. An application more oriented to clinical decision-making can be found in (Oliveira et al., 2014b).

Given the mechanisms provided by Speculative Computation, it is possible to say that Objective 7 (see Table 1) was successfully achieved.

- **A survival prediction model capable of being integrated in a CIG-based CDSS.**

As useful as the knowledge in CIG-based CDSSs may be in guiding the clinical process with well-structured steps, it is static knowledge. There are tasks that require other models of decision-making. Such is the case with survivability prediction.

Although the way of interaction of CIG-based CDSSs is to establish a dialogue, it does not mean that the interlocutor should be a human. This function could very well be taken by another system that offers estimations, predictions, or classifications. With this in mind and considering that the main case study for CompGuide was developed with a CPG for colon cancer, a set of survivability prediction models were developed using ensemble machine learning. The models had a good overall performance and were deployed in a system with a service-oriented architecture in order to facilitate the communication with other systems. It is considered that such a system offers a fitting complement to CIG-based CDSSs.

This contribution allowed for the accomplishment of Objective 8 (see Table 1).

Table 1: Objectives of the doctoral thesis and document sections where they were achieved.

Ojective	Section
Objective 1: Identification and characterization of the main requirements and necessities of CIG systems.	Section 2.1: Development and Implementation of Clinical Guidelines: An Artificial Intelligence Perspective
Objective 2: Identification and analysis of techniques and models stemming from AI and web technologies that can provide the capacities to CIG systems to overcome their challenges.	Section 2.1: Development and Implementation of Clinical Guidelines: An Artificial Intelligence Perspective
Objective 3: Formalization of a comprehensive CIG representation, fit for multiple medical domains and situations in the clinical process.	Section 2.2: Representation of Clinical Practice Guideline Components in OWL Section 2.3: Assessing an Ontology for the Representation of Clinical Protocols in Decision Support Systems
Objecive 4: Development of a library of guideline items with re-usable knowledge patterns that facilitates the encoding of CPGs in a CIG format.	Section 2.2: Representation of Clinical Practice Guideline Components in OWL (further explained in Section 3.1)
Objective 5: Definition of an architecture for the agile deployment of a CIG-based CDSS.	Section 2.4: Decision Support Provided by a Temporally Oriented Health Care Assistant
Objective 6: Suggestion of a method to achieve a higher integration of CIG recommendations in clinical practice.	Section 2.4: Decision Support Provided by a Temporally Oriented Health Care Assistant
Objective 7: Development of a mechanism to manage data-centred uncertainty, namely incomplete information, in the decision points of CIGs.	Section 2.5: A Dynamic Default Revision Mechanism for Speculative Computation
Objective 8: Exploration of dynamic models, for prediction and classification, that offer a complement to CIG recommendations.	Section 2.6: A Mobile and Evolving Tool to Predict Colorectal Cancer Survivability

3.2 VALIDATION OF THE RESEARCH HYPOTHESIS

In order to validate the research hypothesis formulated in Section 1.4, it is necessary to demonstrate how the elements of the CompGuide framework enable the expression of properties such as modularity, flexibility, adaptability, and interactivity, thus allowing the construction of a system with *living guidelines*.

The modularity property is conferred by the ontological definition of CIGs and the assistance in guideline acquisition provided by the CompGuide editor, allowing for the management of multiple knowledge patterns that, once introduced, can be integrated multiple times anywhere in the guideline. Flexibility is also provided by this very ontological definition and the representation primitives which have been shown to accommodate CPGs from different categories and specialities, meaning that the model is adjustable to multiple domains.

On to adaptability, this property is conferred by the mechanisms of Speculative Computation, which allow CIG-based CDSSs to not only reason with incomplete information and build scenarios with the most likely outcome of decisions but also to readjust those scenarios as information becomes known, thus adapting to changes of state. Additionally, the accommodation of situations of uncertainty is further enhanced by models such as the survivability prediction model presented herein.

The interactivity of a CIG-based CDSS is mainly related with the style of communication it uses. The solution proposed herein, the health care assistant, aims to embed CIG advice directly into daily life and provide a set of reminders and notifications to health care professionals that help them to keep track of their tasks and responsibilities. As such, it is considered that this style of communication is more interactive. Perhaps, it can even help to prevent situations of medical error such as the one described in Section 1.2.1, by reminding health care professionals of the time at which actions should take place.

These are features enhance the CIG-based systems and have the potential to meet some of the challenges that clinical practice is currently facing. Taking this into account, it is believed that the research hypothesis was proven.

3.3 DISSEMINATION OF RESULTS

Since the beginning of this PhD project and as outlined in the research methodology of Section 1.6, efforts were directed towards achieving a high level of scientific dissemination through journals, conferences, and other media.

The organization of scientific meetings was also part of these efforts, along with the supervision of master students and lecturing. These activities were undertaken in order to consolidate knowledge and to exchange ideas.

In order to demonstrate the reach of the work developed in this doctoral thesis, the many activities of result dissemination are described in this section

3.3.1 *Other Publications*

Besides the publications included in this thesis, it was possible to achieve other publications in international journals and scientific meetings in the fields of Computer Science, Artificial Intelligence, and Medical Informatics.

International Journals

The work described in this thesis has been documented in international scientific journals through the following list of publications:

- Silva, A., **Oliveira, T**., Neves, J., & Novais, P. (2016). Treating Colon Cancer Survivability Prediction as a Classification Problem. Advances Ins Distributed Computing and Artificial Intelligence Journal, 5, 37–50. http://doi.org/10.14201/ADCAIJ2016513750

- Novais, P., **Oliveira, T.**, & Neves, J. (2016). Moving towards a new paradigm of creation, dissemination, and application of computer-interpretable medical knowledge. Progress in Artificial Intelligence, 5(2), 77–83. http://doi.org/10.1007/s13748-016-0084-2

- Ramos, J., Novais, P., Satoh, K., **Oliveira, T.**, & Neves, J. (2015). Speculative orientation and tracking system. International Journal of Artificial Intelligence, 13(1), 94–119.

- Cardoso, L., Marins, F., Magalhães, R., Marins, N., **Oliveira, T.**, Vicente, H., . . . Neves, J. (2015). Abstract computation in schizophrenia detection through artificial neural network based systems. Scientific World Journal, 2015. http://doi.org/10.1155/2015/467178

- Gomes, M., **Oliveira, T.**, Carneiro, D., Novais, P., & Neves, J. (2014). Studying the Effects of Stress on Negotiation Behavior. Cybernetics and Systems, 45(3), 279–291. http://doi.org/10.1080/01969722.2014.894858

- **Oliveira, T.**, Costa, Â., Neves, J., & Novais, P. (2013). A comprehensive clinical guideline model and a reasoning mechanism for AAL systems. International Journal of Artificial Intelligence, 11(13 A), 57–73.

- **Oliveira, T.**, Novais, P., & Neves, J. (2012). Guideline Formalization and Knowledge Representation for Clinical Decision Support. Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ), 1(2), 1–12. http://doi.org/10.14201/ADCAIJ20121211

Conference Proceedings

The participation in international conferences yielded the following list of publications:

- Gomes, M., **Oliveira, T.**, & Novais, P. (2017). A Speculative Computation Approach for Conflict Styles Assessment with Incomplete Information. In C. Badica, A. El

Fallah Seghrouchni, A. Beynier, D. Camacho, C. Herpson, K. Hindriks,& P. Novais (Eds.), Intelligent Distributed Computing X: Proceedings of the 10th International Symposium on Intelligent Distributed Computing – IDC 2016, Paris, France, October 10-12 2016 (pp. 163–172). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-48829-5_16.

- Silva, A., **Oliveira, T.**, Julian, V., Neves, J., & Novais, P. (2016). A mobile and evolving tool to predict colorectal cancer survivability. In L. Iliadis & I. Maglogiannis (Eds.), IFIP Advances in Information and Communication Technology (Vol. 475, pp. 14–26). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-44944-9_2

- Ramos, J., **Oliveira, T.**, Satoh, K., Neves, J., & Novais, P. (2016). Orientation System Based on Speculative Computation and Trajectory Mining. In J. Bajo, J. M. Escalona, S. Giroux, P. Hoffa-Dabrowska, V. Julián, P. Novais, . . . R. Azambuja-Silveira (Eds.), Highlights of Practical Applications of Scalable Multi-Agent Systems. The PAAMS Collection: International Workshops of PAAMS 2016, Sevilla, Spain, June 1-3, 2016. Proceedings (pp. 250–261). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-39387-2_21

- Silva, A., **Oliveira, T.**, Novais, P., Neves, J., & Leão, P. (2016). Developing an individualized survival prediction model for colon cancer. In H. Lindgren, F. J. De Paz, P. Novais, A. Fernández-Caballero, H. Yoe, A. Jiménez Ram'irez, & G. Villarrubia (Eds.), Advances in Intelligent Systems and Computing (Vol. 476, pp. 87–95). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-40114-0_10

- **Oliveira, T.**, Silva, A., Novais, P., & Neves, J. (2016). A Personal Assistant for Health Care Professionals Based on Clinical Protocols. In Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, & M. Mendonça Teixeira (Eds.), New Advances in Information Systems and Technologies (pp. 845–855). Springer International Publishing. http://doi.org/10.1007/978-3-319-31232-3

- **Oliveira, T.**, Satoh, K., Novais, P., Neves, J., Leão, P., & Hosobe, H. (2016). A reasoning module for distributed clinical decision support systems. In Studies in Computational Intelligence (Vol. 616, pp. 387–397). http://doi.org/10.1007/978-3-319-25017-5_36

- Silva, A., **Oliveira, T.**, Novais, P., & Neves, J. (2015). Representing Temporal Patterns in Computer-Interpretable Clinical Guidelines. In C. Schulz & D. Liew (Eds.), OASIcs - OpenAccess Series in Informatics (Vol. 49, p. 69). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. http://doi.org/10.4230/OASIcs.ICCSW.2015.62

- Ramos, J., **Oliveira, T.**, Novais, P., Neves, J., & Satoh, K. (2015). An alert mechanism for orientation systems based on Speculative computation. In INISTA 2015 - 2015

International Symposium on Innovations in Intelligent SysTems and Applications, Proceedings (pp. 1–18). http://doi.org/10.1109/INISTA.2015.7276741

- **Oliveira, T.**, Satoh, K., Novais, P., & Neves, J. (2014). Speculative Computation with Constraint Processing for the Generation of Clinical Scenarios. In W. Weghorn (Ed.), Proceedings of the 11th International Conference in Applied Computing (pp. 143–150). Porto.

- **Oliveira, T.**, Neves, J., Novais, P., & Satoh, K. (2014). Applying speculative computation to guideline-based decision support systems. In Proceedings - IEEE Symposium on Computer-Based Medical Systems (pp. 42–47).http://doi.org/10.1109/CBMS.2014.32

- **Oliveira, T.**, Leão, P., Novais, P., & Neves, J. (2014). Webifying the Computerized Execution of Clinical Practice Guidelines. In J. Bajo Perez, J. M. Corchado Rodríguez, P. Mathieu, A. Campbell, A. Ortega, E. Adam, ... V. Julián (Eds.), Trends in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection SE - 18 (Vol. 293, pp. 149–156). Springer International Publishing. http://doi.org/10.1007/978-3-319-07476-4_18

- Gomes, M., **Oliveira, T.**, Silva, F., Carneiro, D., & Novais, P. (2014). Establishing the Relationship between Personality Traits and Stress in an Intelligent Environment. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8482 LNAI(PART 2), 378–387. http://doi.org/10.1007/978-3-319-07467-2_40

- **Oliveira, T.**, Neves, J., Barbosa, E., & Novais, P. (2013). Clinical Careflows Aided by Uncertainty Representation Models. In J.-S. Pan, M. Polycarpou, M. Woźniak, A. P. L. F. de Carvalho, H. Quintián, & E. Corchado (Eds.), Hybrid Artificial Intelligent Systems (Vol. 8073, pp. 71–80). Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-40846-5_8

- **Oliveira, T.**, Barbosa, E., Martins, S., Goulart, A., Neves, J., & Novais, P. (2013). A prognosis system for colorectal cancer. In Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems (pp. 481–484). IEEE. http://doi.org/10.1109/CBMS.2013.6627846

- Gomes, M., **Oliveira, T.**, Carneiro, D., Novais, P., & Neves, J. (2013). Extracting Behavioural Patterns from a Negotiation Game. In J. Corchado, J. Bajo, J. Kozlak, P. Pawlewski, J. Molina, V. Julian, ... S. Giroux (Eds.), Highlights on Practical Applications of Agents and Multi-Agent Systems (Vol. 365, pp. 304–315). Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-38061-7_29

- **Oliveira, T.**, Novais, P., & Neves, J. (2013). A Web Platform and a Decision Model for Computer-Interpretable Guidelines. In V. Snášel, A. Abraham, & E. S. Corchado (Eds.), Soft Computing Models in Industrial and Environmental Applications (Vol.

188, pp. 201–210). Springer Berlin Heidelberg.
http://doi.org/10.1007/\978-3-642-32922-7_21

- **Oliveira, T.**, Neves, J., Costa, Â., Novais, P., & Neves, J. (2012). An interpretable guideline model to handle incomplete information. In S. Omatu, J. F. Paz Santana, S. R. González, M. M. Jose, A. M. Bernardos, & J. M. Corchado (Eds.), Advances in Intelligent and Soft Computing (Vol. 151 AISC, pp. 437–444). Springer - Series Advances in Intelligent and Soft Computing. http://doi.org/10.1007/978-3-642-28765-7_52

3.3.2 *Participation and Organization of Events*

Throughout the doctoral programme there were opportunities to participate in international scientific events. These opportunities were important in order to acquire new experiences and exchange ideas. Here is a list of the most important:

- **Imperial College Computing Student Workshop (ICCSW):** The ICCSW aims to provide an international forum for doctoral students to discuss a range of topics that are current in computer science research. The doctoral candidate participated in the 2014 and 2015 editions of the event as an ambassador and reviewer.

- **International Symposium on Intelligent Distributed Computing (IDC):** IDC is a symposium focused on the emergent field of Intelligent Distributed Computing and the development of a new generation of intelligent distributed systems. The doctoral candidate participated in the 2015 edition of the symposium as a member of the organizing committee.

- **Workshop on Decision Making in Dynamic Information Environments (DeMaDIE):** The focus of DeMaDIE is on decision-making in dynamic information environments where uncertainty is typically present. The topics covered by the workshop include default reasoning, argumentation, belief revision in multiagent systems, among others. The doctoral candidate participated in the organizing committee of the first edition of the workshop, held in 2016, at the 15th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'16).

3.3.3 *Invited Presentations*

Other than the publication of scientific articles, research results were also disseminated in invited talks and tutorial demonstrations. These activities occurred during lectures on Intelligent Systems. The list of oral presentations, organized by topic, made in this context is the following:

- **Clinical Decision Support Systems and Computer-Interpretable Guidelines:** presentations to the students of the Masters in Computer Science and the Masters in Biomedical Engineering of the University of Minho.

- **The CompGuide Framework:** presentations to the students of the Masters in Computer Science and the Masters in Biomedical Engineering of the University of Minho.
- **Developing OWL Ontologies with Protégé:** presentations to the students of the Masters in Computer Science and the Masters in Biomedical Engineering of the University of Minho.
- **Speculative Computation with Default Revision:** presentations to the students of the Masters in Computer Science and the Masters in Biomedical Engineering of the University of Minho.
- **Developing Multi-agent Systems with the JAVA Agent DEvelopment Framework (JADE):** presentations to the students of the Masters in Computer Science and the Masters in Biomedical Engineering of the University of Minho.

3.3.4 *Supervision of Students*

During the doctoral programme, the doctoral candidate successfully co-supervised master students at the University of Minho that developed work in the context of this thesis. All these works were supervised by Professor Paulo Jorge Freitas de Oliveira Novais. The list of students and their respective dissertation titles are the following:

- **Tiago Vilas Boas** (2013), Masters in Computer Science, Dissertation: "Automated Creation and Deployment of Clinical Guidelines".
- **António Silva** (2015), Masters in Computer Science, Dissertation: "Representation of Temporal Information in Clinical Decision Support Systems".
- **Ana Silva** (2016), Masters in Biomedical Engineering, Dissertation: "A Survival Prediction Model for Colorectal Cancer Patients".
- **Filipe Gonçalves** (2016), Masters in Computer Science, Dissertation: "Computer-Interpretable Guidelines in Decision Support Systems: Creation and Editing of Clinical Protocols for Automatic Interpretation".

3.3.5 *Research Stays*

In the context of the work developed during the doctoral programme, the candidate spent research periods at two research institutions, the National Institute of Informatics (NII) of Japan and the Valencia University of Technology (UPV).

The research stays at NII consisted of three internships in 2014, 2015, and 2016, under the supervision of Professor Ken Satoh. The purpose of these stays was to study the management of incomplete information in decision-making processes and the development of Speculative Computation as a method to address this issue. It was possible to develop a fruitful collaboration with Professor Satoh that is still producing results. So far, this collaboration has resulted in two publications in international scientific journals and five publications in conference proceedings.

The research stay at UPV was spent at the Grupo de Tecnologia Informática - Inteligencia Artificial (GTI-IA) under the supervision of Professor Vicente Julián. The purpose of the stay was to integrate distributed information sources in a CDSS and account for possible missing information in the inherent decision-making process. Additionally, the secondary objective was to identify common research lines and establish a collaboration between the two research groups. As a result, a paper entitled "A Mobile and Evolving Tool to Predict Colorectal Cancer Survivability" was accepted at the 12th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI 2016).

3.4 FINAL REMARKS AND FUTURE WORK CONSIDERATIONS

With this doctoral thesis it was possible to research some of the most interesting aspects of CDSSs, ranging from representation to reasoning. The work presented herein counted with the precious collaboration of MD Pedro Leão, who provided insightful information about the nature of clinical practice. Unfortunately, it was not possible to test more the proposed solutions, mainly due to inherent difficulties of assessing CDSSs in a real clinical setting. This setting is an extremely sensitive one, where the risk is high and the access is limited and hard to get.

Although it was possible to achieve the above-mentioned results and widen the scope of the research, touching various fields. There were aspects that, although important and mentioned, were not developed. Such is the case of the integration of medical terminologies in the CompGuide ontology, the development of a group decision-making platform for health care professionals supported by CIG advice, and the use of CBR to enhance CIG adaptability. This was due, for the most part, to the necessity of narrowing down the research to what was considered to be achievable.

Emerging trends in CIG research include patient-centric CIG-based CDSSs, which deal with how a patient's personal context and preferences affect decision-making and how a system can accommodate this. The personal context of a specific patient may involve other existing comorbidities or local settings that cannot accommodate certain medical procedures. This line of research aims to help physicians to define care plans that are more accurate and may produce better results.

Another current trend in CIG research is decision support through ubiquitous CIG-based guidance systems that use service-oriented architectures for event-driven modes, through smartphones and web applications. By increasing the capacity of CIG systems to be present in different settings, it becomes possible to deliver decision support at any time and place.

Another unresolved issue in CIG-based CDSSs is to determine how a system should concurrently apply multiple CIGs to patients that have comorbid clinical conditions and sort the conflicts that originate from this situation. This is a common situation in health care, particularly in elderly patients. The difficulty here is how to prioritize recommendations and the criteria to do so.

These emerging research lines reflect the necessity to approximate CIG advice to the place, time, and context of care.

BIBLIOGRAPHY

Babrow, Austin, Kasch, Chris, and Ford, Leigh. "The many meanings of uncertainty in illness: toward a systematic accounting." *Health Communication* 10.1 (1998): 1–23.

Barnett, G. Octo, Cimino, James J., Hupp, Jon A., and Hoffer, Edward P. "DXplain: An evolving diagnostic decision-support system." *Journal of the American Medical Association* 258.1 (1987): 67.

Bates, David W. and Gawande, Atul A. "Improving safety with information technology." *New England Journal of Medicine* 348.25 (2003): 2526–2534.

Benner, Patricia. "From novice to expert." *American Journal of Nursing* March: 402–407.

Benner, Patricia and Tanner, Christine. "How expert nurses use intuition." *The American Journal of Nursing* 87.1 (1987): 23–34.

Bottrighi, Alessio, Terenziani, Paolo, Montani, Stefania, Torchio, Mauro, and Molino, Gianpaolo. "Clinical guidelines contextualization in GLARE." *AMIA Annual Symposium proceedings*. vol. 2006. American Medical Informatics Association, 2006, 860.

Boxwala, Aziz A., Peleg, Mor, Tu, Samson, Ogunyemi, Omolola, Zeng, Qing T., Wang, Dongwen, Patel, Vimla L., Greenes, Robert A., and Shortliffe, Edward H. "GLIF3: A representation format for sharable computer-interpretable clinical practice guidelines." *Journal of Biomedical Informatics* 37.3 (2004): 147–161.

Bright, Tiffani J., Wong, Anthony, Dhurjati, Ravi, and et al. "Effect of clinical decision-support systems: A systematic review." *Annals of Internal Medicine* 157.1 (2012): 29–43.

Bryans, Alison and McIntosh, Jean. "Decision making in community nursing: an analysis of the stages of decision making as they relate to community nursing assessment practice." *Journal of Advanced Nursing* 24.1 (1996): 24–30.

Buckingham, Christopher D. and Adams, Ann. "Classifying clinical decision making: a unifying approach." *Journal of Advanced Nursing* 32.4 (2000): 981–989.

Bucknall, Tracey K. "Critical care nurses' decision-making activities in the natural clinical setting." *Journal of Clinical Nursing* 9.1 (2000): 25–36.

Carnevali, Doris L., Thomas, Mary D., Godson, Linda T., and Waterloo, I. A. *Diagnostic reasoning and treatment decision making in nursing*. Philadelphia, Pa., 1993.

Carroll, John S and Johnson, Eric J. *Decision research: A field guide.* Sage Publications, Inc, 1990.

Catriona, Kennedy. "The decision-making process in a district nursing assessment." *British Journal of Community Nursing* 7.10 (2002): 505–513.

de Clercq, Paul A., Blom, Johannes A., Korsten, Hendrikus H. M., and Hasman, Arie. "Approaches for creating computer-interpretable guidelines that facilitate decision support." *Artificial Intelligence in Medicine* 31.1 (2004): 1–27.

De Dombal, F. T., Leaper, D. J., Staniland, John R., McCann, A. P., and Horrocks, Jane C. "Computer-aided diagnosis of acute abdominal pain." *British Medical Journal* 2.5804 (1972): 9–13.

Eurobarometer. "Patient safety and quality of healthcare Full report." Tech. Rep. October 2009, 2010.

Fox, John, Johns, Nicky, and Rahmanzadeh, Ali. "Disseminating medical knowledge: the PROforma approach." *Artificial Intelligence in Medicine* 14.1-2 (1998): 157–181.

Gardner, Reed, Pryor, Allan, and Warner, Homer R. "The HELP hospital information system: update 1998." *International Journal of Medical Informatics* 54.3 (1999): 169–182.

Grol, Richard. "Successes and failures in the implementation of evidence based guidelines for clinical practice." *Medical Care* 39.8, Supplement II (2001): II46–II54.

Hamm, Robert M. "Clinical intuition and clinical analysis: expertise and the cognitive continuum." *Professional judgment: A reader in clinical decision making* : 78–105.

Hammond, Kenneth R., Hamm, Robert M., Grassia, Janet, and Pearson, Tamra. "Direct comparison of the efficacy of intuitive and analytical cognition in expert judgment." *IEEE Transactions on Systems, Man, and Cybernetics* 17.5 (1987): 753–770.

Hampton, J. R. "Evidence-based medicine, practice variations and clinical freedom." *Journal of Evaluation in Clinical Practice* 3.2 (1997): 123–31.

Han, Kyung-Ja, Kim, Hesook Suzie, Kim, Mae-Ja, Hong, Kyung-Ja, Park, Sungae, Yun, Soon-Nyoung, Song, Misoon, Jung, Yoenyi, Kim, Haewon, Kim, Dong-Oak Debbie, Choi, Heejung, and Kim, Kyungae. "Thinking in clinical nursing practice: A study of critical care nurses' thinking applying the think-aloud, protocol analysis method." *Asian Nursing Research* 1.1 (2007): 68–82.

Han, Paul K. J., Klein, William M. P., and Arora, Neeraj. K. "Varieties of uncertainty in health care: a conceptual taxonomy." *Medical Decision Making* 31.6 (2011): 828–38.

Harbison, Jean. "Clinical decision making in nursing: theoretical perspectives and their relevance to practice." *Journal of Advanced Nursing* 35.1 (2001): 126–133.

Hripcsak, George. "Writing Arden Syntax medical logic modules." *Computers in Biology and Medicine* 24.5 (1994): 331–363.

Hunink, Myriam, Weinstein, Milton C., Wittenberg, Eve, Drummond, Michael F., Pliskin, Joseph S., Wong, John B., and Glasziou, Paul P. *Decision making in health and medicine: integrating evidence and values*. Cambridge University Press, 2014.

Isern, David and Moreno, Antonio. "Computer-based execution of clinical guidelines: A review." *International Journal of Medical Informatics* 77.12 (2008): 787–808.

Kalra, Jawahar. "Medical errors: an introduction to concepts." *Clinical Biochemistry* 37.12 (2004): 1043–51.

Lipshitz, Raanan and Strauss, Orna. "Coping with uncertainty: A naturalistic decision-making analysis." *Organizational Behavior and Human Decision Processes* 69.2 (1997): 149–163.

LUSA. "Ministro manda averiguar casos de doentes sem quimioterapia no Barreiro." 2016. Accessed 2016-12-12.
URL https://www.publico.pt/2016/05/14/sociedade/noticia/hospital-do\-barreiro-abre-inquerito-a-caso-de-doentes-oncologicos-\sem-tratamento-1731972

Makary, Martin A. and Daniel, Michael. "Medical error—the third leading cause of death in the US." *British Medical Journal* 353: 21–39.

Matteson, Peggy and Hawkins, Joellen W. "Concept analysis of decision making." *Nursing Forum* 25.2 (1990): 4–10.

Miller, Morven and Kearney, Nora. "Guidelines for clinical practice: development, dissemination and implementation." *International Journal of Nursing Studies* 41.7 (2004): 813–821.

Mishel, Merle H. "The measurement of uncertainty in illness." *Nursing research* 30.5 (1981): 258–263.

Muir, Nita. "Clinical decision-making : theory and practice." *Nursing standard* 18.36 (2016): 47–52.

Murdoch, Travis B. and Detsky, Allan S. "The inevitable application of big data to health care." *Journal of the American Medical Association* 309.13 (2013): 1351–1352.

Musen, M. A., Shahar, Y., and Shortliffe, E. H. "Clinical decision-support systems." *Biomedical Informatics Computer Applications in Health Care and Biomedicine*. eds. Edward Shortlife and James Cimino. Springer, 2006. 698–736.

News Health Care IT. "IBM Watson pinpoints rare form of leukemia after doctors misdiagnosed patient." 2016. Accessed 2016-12-11.
URL http://www.healthcareitnews.com/news/ibm-watson-pinpoints-rare-form-\leukemia-after-doctors-misdiagnosed-patient

Novais, Paulo, Oliveira, Tiago, and Neves, José. "Moving towards a new paradigm of creation, dissemination, and application of computer-interpretable medical knowledge." *Progress in Artificial Intelligence* 5.2 (2016): 77–83.

O'Brien, Rory. "Overview of Action Research Methodology." 1998. Accessed 2016-12-13. URL http://www.web.ca/{~}robrien/papers/arfinal.html

Offredy, Maxine. "The application of decision making concepts by nurse practitioners in general practice." *Journal of Advanced Nursing* 28.5 (1998): 988–1000.

Oliveira, Tiago, Leão, Pedro, Novais, Paulo, and Neves, José. "Webifying the computerized execution of clinical practice guidelines." vol. 293 of *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2014a. 149–156.

Oliveira, Tiago, Neves, José, Novais, Paulo, and Satoh, Ken. "Applying speculative computation to guideline-based decision support systems." *Proceedings - IEEE Symposium on Computer-Based Medical Systems*. 2014b, 42–47.

Oliveira, Tiago, Silva, António, Neves, José, and Novais, Paulo. "Decision support provided by a temporally oriented health care assistant." *Journal of Medical Systems* 41.1 (2017): 13.

Oxford Online Dictionary. "Definition of decision." 2016. Accessed 2016-12-10. URL https://en.oxforddictionaries.com/definition/decision

Patel, Vimla L., Shortliffe, Edward H., Stefanelli, Mario, Szolovits, Peter, Berthold, Michael R., Bellazzi, Riccardo, and Abu-Hanna, Ameen. "The coming of age of artificial intelligence in medicine." *Artificial Intelligence in Medicine* 46.1 (2016): 5–17.

Pearson, Kay C. "Role of evidence-based medicine and clinical practice guidelines in treatment decisions." *Clinical Therapeutics* 20.SUPPL. C (1998): C80–5.

Peleg, Mor. "Computer-interpretable clinical guidelines: A methodological review." *Journal of Biomedical Informatics* 46.4 (2013): 744–63.

Reason, James. "Human error: models and management." *British Medical Journal* 320.7237 (2000): 768–770.

Russell, Stuart J., Norvig, Peter, Canny, John F., Malik, Jitendra M., and Edwards, Douglas D. *Artificial intelligence: a modern approach*, vol. 2. Prentice Hall Upper Saddle River, 2003.

Sackett, David L. "Evidence-based medicine." *Seminars in Perinatology* 21.1 (1997): 3–5.

Schedlbauer, Angela, Prasad, Vibhore, Mulvaney, Caroline, Phansalkar, Shobha, Stanton, Wendy, Bates, David W., and Avery, Anthony J. "What evidence supports the use of computerized slerts and prompts to improve clinicians prescribing behavior?" *Journal of the American Medical Informatics Association* 16.4 (2009): 531–538.

Scheepers-Hoeks, Anne-Marie J., Grouls, Rene J., Neef, Cees, Ackerman, Eric W., and Korsten, Erik H. "Physicians' responses to clinical decision support on an intensive care unit–comparison of four different alerting methods." *Artificial intelligence in medicine* 59.1 (2013): 33–8.

Seyfang, A., Martínez-Salvador, B., Serban, R., Wittenberg, .J, Miksch, S., Marcos, M., Ten Teije, A., and Rosenbrand, K. "Maintaining formal models of living guidelines efficiently." *Artificial Intelligence in Medicine*. Springer, 2007, 441–445.

Shahar, Yuval, Miksch, Silvia, and Johnson, Peter. "The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines." *Artificial Intelligence in Medicine* 14.1-2 (1998): 29–51.

Shortliffe, Edward H. "Update on Oncocin: A chemotherapy advisor for clinical oncology." *Medical Informatics* 11.1 (1986): 19–21.

———. "The adolescence of AI in medicine: Will the field come of age in the '90s?" *Artificial Intelligence in Medicine* 5.2 (1993): 93–106.

Shortliffe, Edward H., Davis, Randall, Axline, Stanton G., Buchanan, Bruce G., Green, Cordell, and Cohen, Stanley N. "Computer-based consultations in clinical therapeutics: Explanation and rule acquisition capabilities of the MYCIN system." *Computers and Biomedical Research* 8.4 (1975): 303–320.

Sim, Ida, Gorman, Paul, Greenes, Robert A., Haynes, R. Brian, Kaplan, Bonnie, Lehmann, Harold, and Tang, Paul C. "Clinical decision support systems for the practice of evidence-based medicine." *Journal of the American Medical Informatics Association* 8.6 (2001): 527–534.

Smithson, Michael. "Conflict aversion: preference for ambiguity vs conflict in sources and evidence." *Organizational Behavior and Human Decision Processes* 79.3 (1999): 179–198.

Sordo, Margarita, Ogunyemi, Omolola, Boxwala, Aziz A, and Greenes, Robert A. "GELLO: an object-oriented query and expression language for clinical decision support." *AMIA Annual Symposium proceedings*. vol. 2003. American Medical Informatics Association, 2003, 1012.

Ten Teije, A., Miksch, S., and Lucas, P. "Computer-based medical guidelines and protocols: A primer and current trends." *Computerbased Medical Guidelines and Protocols A Primer and Current Trends*. vol. 139. 2008. 22–43.

Thompson, Carl and Dowding, Dawn. *Essential decision making and clinical judgement for nurses*. Churchill Livingstone, 2009.

Tu, Samson W., Campbell, James R., Glasgow, Julie, Nyman, Mark A., McClure, Robert, McClay, James, Parker, Craig, Hrabak, Karen M., Berg, David, Weida, Tony, Mansfield, James G., Musen, Mark A., and Abarbanel, Robert M. *The SAGE Guideline Model: achievements and overview.*, vol. 14. 2007.

Vincent, Charles, Neale, Graham, and Woloshynowych, Maria. "Adverse events in British hospitals: preliminary retrospective record review." *British Medical Journal* 322.7285 (2001): 517–519.

Watson, Steve. "An exploratory study into a methodology for the examination of decision making by nurses in the clinical area." *Journal of Advanced Nursing* 20.2 (1994): 351–360.

Wennberg, J.E. “Unwanted variations in the rules of practice.” *JAMA* 265.10 (1991): 1306–1307.

Wilson, Ross, Runciman, William B., Gibberd, Robert W., Harrison, Bernardette T., Newby, Liza, and Hamilton, John D. “The quality in Australian health care study.” *Medical Journal of Australia* 163.9 (1995): 458–471.

Woolf, Steven H., Grol, Richard, Hutchinson, Allen, Eccles, Martin, and Grimshaw, Jeremy. “Potential benefits, limitations, and harms of clinical guidelines.” *British Medical Journal* 318.7182 (1999): 527–530.

World Health Organization. “Patient safety: data and statistics.” 2016. Accessed 2016-12-12.
URL http://www.euro.who.int/en/health-topics/Health-systems/patient-safety/data-and-statistics