Universidade do Minho
Escola de Engenharia

António José Linhares da Silva

A Decision Support System Based on Guidelines with Conflict Resolution Features

September, 2021

A Decision Support System Based on Guidelines with Conflict Resolution Features

António José Linhares da Silva

UMinho | 2021

Universidade do Minho
Escola de Engenharia

António José Linhares da Silva

A Decision Support System Based on Guidelines with Conflict Resolution Features

Doctoral Thesis

Doctoral Program in Informatics

Work supervised by

Paulo Jorge Freitas de Oliveira Novais

Tiago José Martins Oliveira

September, 2021

COPYRIGHT AND TERMS OF USE OF THIS WORK BY A THIRD PARTY

License granted to the users of this work

Acknowledgements

"Just as electricity transformed almost everything 100 years ago, today I actually have a hard time thinking of an industry that I don't think AI will transform in the next several years." - Andrew Ng

There are obstacles in life that nobody is unaware of, and in these four years, the path has not always been easy. During this period, there were better times and worse times, but above all, I keep in my heart the good times, the moments of learning both personally and academically.

Thanks to my parents', the difficulties were surpassed for their unconditional support and the education that they gave me. There are no words that can thank all the support provided, just the pleasure of seeing me with a doctorate and by realising that all the efforts and sacrifices made were not in vain. To my brother for his unconditional support, for his strength and affection. To all my family for all the help and strength provided. There is also a word to my uncle Luís and grandfather Francisco, who are no longer physically by my side, but it was for them too that I had the strength to achieve this goal.

I learned to be more responsible in this house of knowledge and face problems with determination and courage, but I mainly learned never to give up. Therefore, I would like to thank the University of Minho, for making me grow at all levels, during my master's and doctorate. A very special thanks to Professor Paulo Novais and Tiago Oliveira for the confidence placed on me, for the availability and patience to guide me during my PhD. Also, I would like to thank all other professors at the University of Minho, who have taught me a lot over the years. Of course, I could not forget the first house that saw me grow, and, thus, a huge thanks to ISEP and all the teachers for making me an Engineer.

Also, I have much to thank Ms Margarida by the emotional support, since in many times brought me positive energy which enabled me not to be discouraged in critical moments.

I want to thank the ISLab members and all the friends that I made there, for the excellent work environment, the availability, and the patience in transmitting the knowledge, thus, providing an enrichment experience in this academic path.

Finally, I would like to thank all my friends for the advice given, for the support and the trust transmitted over these years.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

___Braga___,

(Location)

Assinado por : **ANTÓNIO JOSÉ LINHARES DA SILVA**
Num. de Identificação: 13906897
Data: 2021.09.06 12:01:07+01'00'

CHAVE MÓVEL

(António José Linhares da Silva)

Resumo

Um Sistema de Apoio à Decisão Baseado em Protocolos com Recursos de Resolução de Conflitos

Atualmente no setor da saúde, existe uma crescente necessidade de sistemas que sejam mais interativos, pervasivos eficazes no suporte à decisão clínica. Além disso, faltam sistemas capazes de fornecer suporte à decisão clínica, através de recomendações especificas para cada paciente e que se baseiam em versões de protocolos clínicos para interpretação automática. Alguns desafios à modelação de Computer-Interpretable Guidelines (CIGs) surgiram, dificultando a representação computacional de protocolos clínicos e a sua respetiva integração em Sistemas de Apoio à Decisão Clínica . Esses desafios são a modelação, reutilização e combinação de conhecimento, representação de informação temporal, criação, edição e execução de protocolos clínicos e a identificação automática de conflitos e interações entre recomendações bem como a sua mitigação.

Embora as soluções atuais ofereçam ferramentas para criar e executar protocolos clínicos, elas carecem em funcionalidades como agendamento e gestão temporal de protocolos clínicos, combinação de protocolos clínicos, identificação automática e mitigação de conflitos entre diferentes protocolos clínicos e ferramentas *user-friendly* para criação e edição de CIGs. Outros sistemas, apesar de apresentarem várias funcionalidades e ferramentas, são difíceis de usar, com interfaces complicadas e não intuitivas, o que os torna menos adequados para o uso diário em um ambiente tão complicado que é uma unidade de saúde.

Como meio de resolver esta questão, o presente trabalho de doutoramento propõe um método que identifica e mitiga automaticamente os potenciais conflitos ou interações comuns que podem ocorrer ao combinar os diferentes protocolos para pacientes multimórbidos. Além disso, esta abordagem oferece a possibilidade de gerir a criação e edição de CIGs, bem como fornece mecanismos para integrar e codificar recomendações de CIG na rotina diária dos profissionais de saúde.

Palavras-chave: Análise de Decisão Multicritério, Argumentação Computacional, Computer-Interpretable Guidelines, Ontologias, Suporte à Decisão Clínica.

Abstract

A Decision Support System Based on Guidelines with Conflict Resolution Features

Currently, in the health sector, there is a growing need for systems that are more interactive, pervasive, and effective in clinical decision support. Also, there is a lack of systems capable of providing decision support through patient-specific recommendations based on Clinical Practice Guideline (CPG) versions for automatic interpretation. Some challenges to CIGs modelling have arisen, making it difficult to represent clinical protocols computationally and integrate them in Clinical Decision Support Systems (CDSSs). These challenges are modelling, reusing and combining the knowledge, representation of temporal information, creation, editing, and execution of CPGs and the automatical identification of recommendation conflicts and interactions and their respective mitigation.

Although current solutions offer tools to create and execute CPGs, they lack in functionalities such as scheduling and temporal management, the combination of knowledge from multiple CPGs, automatical identification and mitigation of conflicts among different CPGs, and user-friendly tools to create and editing of CIGs. Others, despite being filled with many functionalities and tools, are difficult to use, with complicated and non-intuitive interfaces, which makes them less suitable for daily use in such a complicated environment that is a health facility.

As a means to solve this issue, this doctoral work proposes a method that automatically identifies and mitigates the common potential conflicts or interactions that can happen when merging CPGs for multimorbid patients. Furthermore, this approach offers the possibility of managing the creation and editing of CIGs as well as provide mechanisms for integrating and scheduling CIG recommendations in the daily routine of health care professionals.

Keywords: Clinical Decision Support, Computational Argumentation, Computer-Interpretable Guidelines, Multiple Criteria Decision Analysis, Ontologies.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AHP Analytic Hierarchy Process

AI Artificial Intelligence

AIIM Artificial Intelligence in Medicine

API Application Programming Interface

ASTM American Society for Testing and Materials

ATC-code Anatomical Therapeutic Chemical Classification System

BRA Benefit-risk Assessment

CDSS Clinical Decision Support System

CIG Computer-Interpretable Guideline

CLP Constraint Logic Programming

CPG Clinical Practice Guideline

DBMS Database Management System

DeGeL Digital Electronic Guideline Library

DTS Distributed Terminology System

EMA European Medicines Agency

EMR Electronic medical record

GEE Guideline Execution Engine

GLARE Guideline Acquisition, Representation and Execution

GLEE Guideline Execution Engine

GLIF Guideline Interchange Format

GLIF3 Guideline Interchange Format version 3

HL7 Health Level Seven

IJMI International Journal of Medical Informatics

JAMIA Journal of the American Medical Informatics Association

JBI Journal of Biomedical Informatics

MCDA Multiple-criteria Decision Analysis

MIIM Methods of Information in Medicine

MLM Medical Logic Module

NCCN National Comprehensive Cancer Network

OKBC Open Knowledge Base Connectivity

OWL Ontology Web Language

PAL Protégé Axiom Language

RBC Rule-based Combinations

SAGE Standards-Based Sharable Active Guideline Environment

SDM Shared Decision-Making

STP Simple Temporal Problem

TM4I Transition-based Model for automatic Inference of Interactions

TNM Task Network Model

XML eXtensible Markup Language

Chapter 1

Introduction

The present doctoral thesis, developed within the Doctoral Program in Informatics at the University of Minho, has the theme: ”A Decision Support System Based on Guidelines with Conflict Resolution Features”.

This doctoral thesis covers areas of Artificial Intelligence (AI) which have a more application-oriented approach, such as eHealth, Medical Informatics, CIGs, and CDSSs.

This chapter, section 1.1, provides a theoretical background of the work and the motivation that underlies it. Sections 1.2 and 1.3 clarify some aspects regarding the functions and the importance of CDSSs and the role that CPGs play in their development. Section 1.4 points out the challenges to the representation of CPGs as CIGs. Section 1.5 defines the research hypothesis. The motto for the work is treated in section 1.6 in the form of a set of research questions that were later transformed into objectives. Section 1.7 describes the methodology for achieving the proposed objectives, the phases are briefly described, and the research design is set. Lastly, section 1.8 describes the document’s organisation and the topics covered in each chapter.

1.1 Background and Motivation

The subject of this doctoral thesis involves important scientific areas, so it is necessary to frame the problems that will be addressed in the different scientific areas that they touch.

The dominant area of this work is eHealth. eHealth is a recent term that describes the practice of health care supported by electronic processes [1]. The research in this area aims to develop electronic/digital methods in health and, as such, it is considered that it is a sub-area of Medical Informatics [2] [3].

Another area of great practical interest to this work is AI since it is an area of computational research dedicated to finding computational methods or devices that possess or multiply the intellectual capacity of the human being to solve problems, think or reason, broadly. It can also be defined, according to Russell et al. [4], as the field of computer science that deals with intelligent behaviour or the study of how to make computers perform tasks that are currently better performed by human beings [5].

In this context, AI has something to say about the techniques used in eHealth technologies by improving its implementations, which can be achieved through decision support systems.

CDSSs are specialist systems designed to assist physicians and other health care professionals performing clinical tasks, such as determining the diagnosis based on patient data [6]. This aspect is evidenced in the definition proposed by Musen et al. in [7] that CDSSs link clinical observations to medical knowledge in a way that positively influences the choices of health professionals and contributes to improved health care. However, as the definition entails, it is necessary to support the representation of medical knowledge, which allows for the automatic crossing of medical knowledge and observations. This support can exist in several forms, and one of the most used is the algorithms based on clinical protocols in formats that enable their automatic interpretation.

1.2 Clinical Decision Support Systems

The research to create artificially intelligent computer systems has been one of the most ambitious and, not surprisingly, controversial activities in the field of computer science [8]. From early on, the researchers in health were also attracted by the potential that this technology can have in medicine [9]. Thus, the initial focus of AI in Medicine was the development of systems capable of performing the diagnosis and provide therapy recommendations [10].

By definition, CDSSs are active knowledge systems that use patient data items to generate case-specific recommendations. These systems include a medical knowledge base, support for patient data, and an inference engine [7]. A CDSS is any computer program designed to help health care professionals to make clinical decisions. In a way, any computer system working with clinical data or medical knowledge is intended to provide decision support [6]. Therefore, it is important to consider the four types of functions that these systems can perform [11]:

- **Administrative**: The administrative function seeks to support clinical codification and documentation, authorisation of procedures and references;
- **Management of Clinical Complexity**: The function of management of clinical complexity aims to manage patients in research and treatment protocols as well as follow-up of prescriptions, follow-up references, and preventive care;
- **Cost Control**: The purpose of cost control is to monitor drug prescriptions, avoiding duplication or unnecessary tests;
- **Decision Support**: This function entails providing support to the processes of diagnosis and planning of the clinical treatments, as well as promoting the use of the best clinical practices.

After a brief description of CDSS functions, it is also important to address the general model of these systems. The general model of a CDSS can be seen in Figure 1.

As shown in Figure 1, the system inputs consist of signs, symptoms, laboratory tests, and others, whereas the outputs, include diagnostic and therapeutic recommendations. There are two core components: a knowledge base and an inference mechanism. The knowledge base is a structured set of specialised medical knowledge, and the inference mechanism is a set of computational algorithms used to process clinical signs, symptoms, and results of laboratory tests concerning the knowledge base.

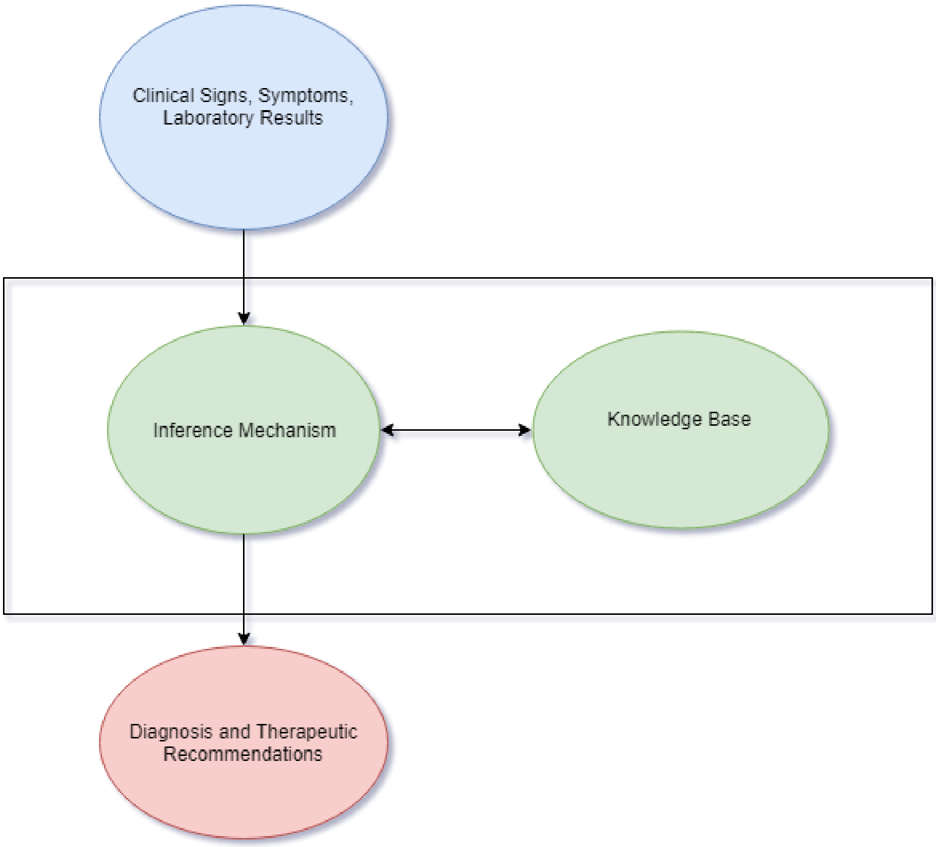

Clinical Signs, Symptoms, Laboratory Results
Inference Mechanism
Knowledge Base
Diagnosis and Therapeutic Recommendations

Figure 1: General Model of Clinical Decision Support Systems.

According to the general model of a CDSS described in Figure 1, the users interact with the CDSS iteratively by selectively introducing the information and using the output recommendations of the CDSS to support the formulation of diagnoses and decision-making in therapeutic processes [12].

The research about the integration of CIGs in CDSSs has grown over the years. At the same time, the challenges CDSSs were already faced with became ever more prominent. The most important of these challenges are the following [13]:

- **Improve the human-computer interface** - It is always necessary to involve the paradigms of the human-computer interface, which includes providing recommendations to support the decision-making. The continuous workflow of health professionals should be supported, which means reminding and monitoring;

- **Prioritise and filter recommendations for the user** - It is important to provide a robust, reliable and evidence-based system. These aspects represent the main challenge; therefore, it is relevant to take into account the influences and competing values that impact clinical decision making. An additional challenge is the reduction of the number of recommendations generated by this type of system. This has the potential to reduce "alert fatigue", which is a phenomenon that health professionals often experience;

- **Combine recommendations from different CPGs** - clinical treatments often ignore the fact that many elderly patients suffer from multiple diseases and take various medications. The challenge here is to create mechanisms to identify and eliminate recommendations that are contraindicated, discordant, or mutually exclusive. Hence, this type of system must present various types of clinical guideline recommendations.

Various technologies are used in CDSSs, such as machine-learning, ontologies, and others. Some CDSSs can learn from data, but currently, most CDSS are rule-based [14]. It is suggested that this is caused by the limited availability of structured data and limited dissemination of algorithms for pattern discovery among clinicians. Besides, rule-based CDSSs can explain the reasoning behind specific advice. Some research projects that are examples of these systems will be presented below.

The KARDIO solution is an automatic learning system that was developed to interpret electrocardiograms to diagnose cardiac arrhythmias. Given a set of clinical cases that serve as examples, the system can produce a systematic description of these clinical characteristics that characterise certain health conditions. This knowledge can be expressed in the form of simple rules or often as a decision tree [8].

The HELP system is an integrated hospital information system developed at LDS Hospital in Salt Lake City [6]. This system supports the routine applications of a hospital information system, including admissions and medical discharge management and order entries, and provides decision support functions. The decision support provides health professionals with alerts and reminders, data interpretation and patient diagnostic environments, patient management suggestions and clinical protocols [8].

DXplain is a CDSS developed at the computer science lab of the Massachusetts General Hospital and used to assist diagnosis. Based on a set of clinical outcomes, including signs, symptoms and laboratory data, the system produces a ranked list of diagnoses that may explain (or be associated with) the clinical manifestations, justifying each of the diseases that may be considered [8].

A different approach to decision support has been incorporated into MYCIN, a consultation system that focuses on the proper management of patients who have infections to the detriment of the diagnostic tasks [6]. The MYCIN system was designed to recommend the treatment of certain blood infections based on IF-THEN rules. The knowledge about infectious diseases in MYCIN is represented as production rules, each containing a knowledge package derived from discussions between health professionals.

Zynx Health is a CDSS developed by the *Hearst Corporation*, which helps hospitals improve patient outcomes and clinical monitoring. The evidence-based tools in this system provide information to health professionals and workflow suggestions, encouraging collaboration between all parties to improve clinical outcomes.

The Cerner system [15], owned by *Cerner Corporation*, uses a set of evidence-based standards and criteria to provide healthcare professionals with reliable guidance to ensure that patients receive the most appropriate treatment for their needs. This system also supports clinical decisions for a diverse range of health services, such as in the field of radiology. Also, health professionals are provided with information on the clinical workflow to allow more precise prescriptions to improve patient care.

The present thesis work focuses on CDSSs that provide decision support through patient-specific recommendations based on CPG versions for automatic interpretation, and CDSSs capable of providing a workflow for creation and execution of CPGs. Additionally, part of the work in this thesis aims to understand how the knowledge of clinical protocols can be modelled, reused and combined so that a system provides features that increase its effectiveness, interactivity and pervasiveness.

1.3 Clinical Practice Guidelines

CPGs are documents developed in a systematic way that aim to improve the quality of health care, reduce variations in medical practice, and reduce health care costs. CPGs help healthcare professionals gather, evaluate and implement best practices in medical care [16]. To be effective, they must be integrated into the care flow and provide patient-specific advice when and where it is needed [17]. These documents accumulate and reflect knowledge on how to best diagnose and treat diseases in the form of a list of clinical recommendations.

They often consist of sections on epidemiology, diagnosis, treatment recommendations and underlying scientific evidence. However, due to a variety of barriers, such as lack of knowledge and lack of agreement with recommendations, health care professionals do not always adhere to CPGs [18] [19].

A CPG can act as a guide to assist the healthcare professional. An example of its application can be verified when a healthcare professional needs to review the administration of a given drug to a patient in a given case. CPGs allow health professionals access to the treatment plan. Additionally, they provide tasks for the monitoring of the patient's health condition [17].

The formalisation of CPGs for automatic interpretation makes it possible to develop decision support systems based on CIGs that offer a better possibility of affecting clinical behaviour concerning narrative documents of the corresponding text versions. The CIGs are one of the possible supports for medical knowledge in decision support systems. Through the formalisation in CDSSs, a new range of operations can be performed with the knowledge CPGs enclose. Such includes automated reasoning for the generation of recommendations, automatic identification of conflicts between CIGs, consistency checking within the same CIG and across different CIGs, and merging CIG knowledge with contextual information such as patient and physician preferences or available health care resources. There are some models of representation, such as Arden Syntax [20], Guideline Interchange Format (GLIF) [21], Asbru [22], PROforma [23] and the Standards-Based Sharable Active Guideline Environment (SAGE) [24].

Except for the Arden Syntax, which is a model that specifically encodes small fragments of medical knowledge in the form of rules, the majority of the models use task networks to represent and expose knowledge of the clinical protocols. The task network model seems to be the one that best fits the information conveyed by clinical protocols, allowing a clear separation between the procedural knowledge, i.e. the knowledge about the relative order of the recommendations, and the medical knowledge.

A model is usually accompanied by an execution mechanism that is responsible for interpreting the clinical constraints and temporal constraints placed on tasks. Tools such as Arezzo™ (for Proforma), the Digital Electronic Guideline Library (DeGeL) for Asbru, the Guideline Execution Engine (GLEE) for Guideline Interchange Format version 3 (GLIF3), and SAGE Desktop are used to provide recommendations interactively and for the storage of clinical protocol executions represented in their respective models [25].

1.4 Challenges to Modelling CPGs

To promote adherence to CPGs and support healthcare professionals in providing care according to best practices, CDSSs are often implemented. These may present relevant medical knowledge, guide workflow, answer questions, retrieve information, perform calculations or group elements. The formalisation of CPGs as CIGs aims to facilitate the integration into CDSSs. However, a variety of challenges difficult its formalisation.

Most of the current CPGs were not designed to be digitally represented and computer-interpretable since they imply complex instructions and the manipulation of too many variables, which lead to non-deterministic and complex algorithms [26][27].

The vocabulary used in the documents is often evasive, featuring words to quantify measures rather than numerical limits, and the criteria in the decision points are not always explicit and do not indicate what to do. The lack of precision of concepts gives rise to ambiguity and gaps in knowledge, which computers cannot handle [27]. The greater the simplicity and assertiveness of a CPG, the easier it is to adapt it to the CIG format. The ambiguity concerning these documents may lead to some issues, namely: the misunderstanding of medical terms (semantic ambiguity), conflicting instructions (pragmatic ambiguity), and the incorrect structure of statements (syntactic ambiguity) [28].

The modelling of the workflow structure of CPGs is also a challenging task. Due to complex execution structures, the task order can be challenging to process. Some tasks can be executed sequentially, others as alternatives, and others in parallel, leading to more complexity to control the execution of the task. Another challenge appears in scheduling tasks since it involves the processing of a wide variety of complex temporal constraints, which directly affect the task enactment times.

Another problem is that CPGs have a single-disease focus and tend not to account for the effect of recommendations for one disease on other diseases. As a result, multimorbid patients often receive multiple disease-specific treatment plans. The result is a concurrent execution of treatment recommendations from different CPGs, which may cause conflicts. Moreover, the application of multiple CPGs individually can result in complex multiple drug regimens (polypharmacy) with the potential for harmful combinations of drugs [29]. Thus, it is crucial to provide alternative solutions that adequately address these conflicts. However, it is not always possible to offer alternative solutions, either because there are no alternative drugs or because alternative prescriptions also lead to drug-drug conflicts.

1.5 Research Hypothesis

The work disclosed in this thesis intends to provide solutions to address the challenges described in section 1.4. It seems clear that there is a variety of obstacles to the adoption of CPGs in daily clinical practice, and even current CIG-based CDSSs do not adequately address them. Thus, it is crucial to approximate CPGs to the concept of *living guidelines*, which includes granting effectiveness and interactivity properties to CPGs. CIGs are considered to be the best approach to the concept of *living guidelines*, which captures statements for clinical decision support that are dynamic - in the sense that they are capable of evolving and providing advice based on the latest evidence - and interactive [30].

The goal of this thesis is to increase the effectiveness, interactivity and pervasiveness of the CIG systems by deploying CIGs in CDSSs, enabling new information and communication services that transparently support physicians in their duties. This system should include alerts and reminders that assist the health care professionals in taking control of the whole clinical process, intelligently and automatically summarising all of the patient's data and allowing the scheduling and temporal management of clinical recommendations. Additionally, these systems should be capable of identifying and mitigating conflicts and interactions among different CPGs. To enable the expression of the interactivity property, the system should provide features for supporting care, medical decisions and ease the burden of health professionals in keeping track of their clinical activities. On to the effectiveness property, the system should allow representing all aspects of the clinical domain and clinical practice. Regarding pervasiveness, the system should always be available during clinical practice.

To conclude, the research hypothesis is as follows:

The proposed framework for CIG-based CDSSs enables the expression of properties such as effectiveness, interactivity and pervasiveness.

1.6 Objectives

This doctoral thesis work presents the theme: "A Clinical Decision Support System based on Guidelines with Conflict Resolution Features". The present research proposal aims to develop a model that allows the execution of CPGs to provide clinical recommendations for patient follow-up. To achieve this objective, it is important to provide the correct information at the proper time in a readable format - that corresponds to three logical parts of CDSS:

1. **Clinical Knowledge Representation**: The CIG model must provide knowledge primitives to represent workflows of recommendations, conditions about the state of the patient, temporal restrictions such as durations, waiting times and periodicities. The CIG model must support the optimisation of the workflow of recommendations to multimorbid patients by deliberately addressing patient preferences and anticipating added complexity.

2. **Decision Making**: The users should be provided with an up-to-date overview of relevant information, including (parts of) CPGs. The functions of this component include checking the ordering and temporal constraints placed on tasks and reusing and combining knowledge from multiple guidelines.

3. **Focusing attention**: Provide alerts and reminders to the user for specific conflicts that pose a high risk of impairing the patient state. Such alerts are task enactment times and steps to collect information about the patient, such as the outcomes of clinical tasks, monitorisation of all patient parameters and clinical processes.

Thus, the research questions that guided the execution of the work are:

A. What are the limitations of the CIG models when looking for ways to reuse, combine, and reason over existing clinical knowledge?

B. What are the temporal aspects of the execution of clinical tasks that must be improved in the current CIG models?

C. How to implement and integrate the temporal aspects into the CIG models?

D. What aspects could be improved or applied in CIG tools?

E. What aspect should be integrated to provide a tool that allows a workflow for the creation and execution of CPGs?

F. In case of existing conflicts and interactions between concurrently executed clinical recommendations and non-existing of alternative solutions to solve them, how to mitigate the recommendation conflicts and how to properly provide an assessment of existing solutions?

The research questions previously specified allowed to formulate the following objectives to be achieved:

A. Analysis of the current state of the art models for the deployment of CIGs in CDSSs and identification of the main challenges to modelling CPGs as CIGs;

B. Identification and characterisation of the requirements of CIG systems that adequately address the identified limitations;

C. Formalisation of a CIG model that provides a comprehensive representation of multiple clinical domains and situations of clinical practice;

D. Definition of architecture for the deployment of CIGs in CDSSs and design an execution engine that handles the workflow of the tasks in the CPGs;

E. Definition of a formalism for the automatic identification and mitigation of the common potential conflicts and interactions that can happen when merging CIGs;

F. Design a platform that permits the visualisation of clinical recommendations and monitors the progress of the clinical process.

1.7 Research Methodology

To developed this doctoral work, the Science Research methodology was employed. In this doctoral work, we combine it with the Hypothetical-Deductive scientific method. It can be distinguished from normal problem-solving by emphasising scientific investigation, which consists of researching and developing a solution for a problem. Moreover, it allows the combination of research and refinement of the hypotheses based on the reviewed literature, which leads the researcher to think and reflect on the implications of the developed theories. Firstly, the researcher studies the problem, and his actions are validated in efficacy, efficiency, and utility by the scientific community. In an initial phase, the researcher starts with a research hypothesis, based on a thorough review of state of the art, in this case of CIG-based CDSSs. This step results in a report of available resources and features. According to what was exposed in Section 1.4, the hypothesis is defined in Section 1.5. Afterwards, a set of objectives is established to prove or refute the research hypothesis. Accordingly, the objectives of the thesis are outlined in Section 1.6. 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 Science Research were applied to each one of these aspects separately. There are, in total, five steps modelled in Science Research. The methodology applied encompasses the following stages of development:

- **Diagnosing**: Definition of the problem and its characteristics, based on the reviewed literature. The problem definition aims to gather information about all its features. Afterwards, a research hypothesis is defined as a solution to the identified problem;

- **Design solutions**: Design and propose solutions specimens according to the information gathered in previous stages and definition of objectives;

- **Evaluation**: Implementation of a prototype that fulfils the previously defined requirements and observation of the solution behaviour to verify its efficacy in problem-solving;
- **Validation**: Analysis, validation and correction of the prototype based on the obtained results. This allows to make proper conclusions about the research hypothesis;
- **Learning dissemination**: Dissemination of knowledge and results obtained in the scientific community.

The first steps are diagnosing the problem, researching state of the art for possible solutions, and updating the objectives of the work. The next step encompasses the design of the solution that fulfils the objectives defined and then the implementation of the proposed solution. The target population of this doctoral work are health care professionals. Regarding data acquisition, we collected information from multiple sources, namely RxNorm API, National Comprehensive Cancer Network (NCCN) Clinical Practice Guideline for Prostate Cancer [31], IDF Clinical Practice Recommendations for managing Type 2 Diabetes [32], NCCN Clinical Practice Guideline for Colon Cancer [33]. To support software development, we use the SCRUM methodology [34]. It is aimed at solving long product development, focusing on a set of sprints. In each sprint, we established a set of goals to be performed. When the above-mentioned tasks are completed, an evaluation of the work is done based on the scientific community feedback.

1.8 Document Structure

This thesis is presented as a compilation of publications. The present thesis is structured in four chapters: Introduction (Chapter 1), Computer-Interpretable Guideline Models (Chapter 2), Publications Composing the Doctoral Thesis (Chapter 3), and Conclusions (Chapter 4). Next, we will present a brief description of each chapter.

The current chapter is the *Introduction*. It gives a background of the work, an introduction to the main concepts and a presentation of the motivation and challenges of the doctoral work. Based on these limitations and challenges, the research hypothesis and the objectives for the thesis are defined. Afterwards, the research methodology is outlined. A short description of the doctoral thesis structure is also given.

The next chapter is the *Computer-Interpretable Guideline Models*, where the CIG projects are studied, and the challenges in these models are analysed and discussed. Some topics such as CDSSs, CIGs, CIG modelling languages, CIG Execution Engines, Models of Temporal Reasoning, Clinical Knowledge Representation, Combining CPGs, CIG Tools, Multimorbidity and MCDA are addressed.

The *Publications Composing the Doctoral Thesis* chapter is where we select relevant publications to integrate the doctoral thesis. These publications were selected since they are extended paper versions and provide a comprehensive description of the work developed. In these articles, we address topics, such as CDSSs, CIGs, Models of Temporal Reasoning, Clinical Knowledge Representation, CIG Tools, Combining CPGs, Computational argumentation, Medical reasoning, Multimorbidity and MCDA. Also, we provide a table of relevant information about the featured publication at the beginning of each section. This chapter is divided into the following sections:

- **Decision Support Provided by a Temporally Oriented Health Care Assistant**

  The publication featured in this section presents the functionalities that enable health care professionals to track and follow their patients, schedule clinical procedures that should be performed, and manage the temporal constraints placed on those procedures. It provides details about the temporal representation model and the basic structure of a CPG in the CompGuide model. Also, it provides details about the tool that builds an agenda for the health care professional with the activities that he has to perform, and how the execution engine schedules the execution of clinical tasks and keeps track of their execution while trying to promote the fulfilment of their temporal constraints.

- **A system for the management of clinical tasks throughout clinical process with notification features**

  The paper included in this section presents the details about the integration of functionalities supporting care in the daily life of health care professionals. It offers information on how we map clinical recommendations, with their respective temporal constraints, to an agenda of activities for a health care professional to perform. Although it presents some work previously disseminated, it presents more implementation details about the applications for supporting care, namely the CompGuide architecture, their RESTfull web services, the features provided by the Personal Assistant Web Application and Health Care Assistant Mobile Application and the integration of the Google calendar in previous applications.

- **Enhancing Decision Making By Providing A Unified System For CIG Management**

  This paper presents a comprehensive architecture for the deployment of CIGs in CDSSs, featuring components that allow: the creation and manipulation of clinical practice guideline knowledge elements, execution of CIGs with the temporal verification of clinical tasks, and drug conflict identification and resolution. In addition, it presents details about the plugin that allows the management of CIGs by allowing their creation and editing, the CIG execution engine that integrates features that allow the automatic identification of drug conflicts and their resolution. Also, it presents details on how we integrate all these components in a unified workflow for the acquisition and customisation of CIGs and their respective deployment.

- **Providing Alternative Measures for Addressing Adverse Drug-drug Interactions**

  The paper featured in this section presents an MCDA approach that fulfils a gap of the former version of the CompGuide framework, which is related to the fact that in most cases, it is not possible to provide alternative drugs to solve the drug-drug conflicts. It presents the methods for assessing and integrating multiple criteria and comparing and assessing different decision alternatives. It thus helps the health professionals to make well-informed clinical decisions, even when there are no alternative drugs to address the identified drug-drug conflicts. Due to space constraints, it was impossible to provide all the details of this comprehensive model in this article, so, in appendix B, we provide more information about the MCDA approach.

- **Mapping a Clinical Case Description to an Argumentation Framework: A Preliminary Assessment**

  This article provides another method for mapping the clinical cases using argumentation theory to an argumentation framework. For this purpose presents a multimorbidity clinical case description extracted from the MIMIC database and uses the ASPIC+G argumentation framework for instantiating it. This framework consists of another formal process to computationally represent clinical actions and their respective components and reasoning in complex scenarios such as multimorbidity clinical cases.

Finally, the last chapter of this thesis summarises the work accomplished, the description of the contributions and how they answer the initial objectives. Also, we present how the research hypothesis is validated. Moreover, we enumerate the activities undertaken for the dissemination of results. Perspectives on future work are also mentioned.

Chapter 2

Computer-Interpretable Guideline Models

This chapter provides an analysis of the works available in the literature and the various applications that currently exist.

The methodology used to elaborate the research on the topics covered in this chapter consisted mainly of the analysis of conference articles and journals available in the databases of Google Scholar, Science Direct, Pubmed, and others. Among the terms researched, it is possible to highlight: CDSSs, CIGs, CIG modelling languages, CIG Execution Engines, Models of Temporal Reasoning, Clinical Knowledge Representation, CIG Tools, Combining CPGs and Multimorbidity.

The objective of this analysis is not only to identify the relevant points of the various solutions that could be incorporated in the solution presented, but also to reflect on the most important aspects of the domain of this doctoral project.

In the following sections, the main CIG approaches will be studied, and some challenges that currently these models face will be addressed. These challenges will be further analysed to identify the literature solutions.

Figure 2 shows the number of papers used as a reference in this document, published in each category from each journal. This figure intends to show the importance of each topic addressed in this chapter as well as its relevance to the scientific community. Due to the high number of referenced articles, only journal articles were selected. The selection of journals follows the methodology described in [35], which consists in selecting five prominent journals in the field of medical informatics: Journal of the American Medical Informatics Association (JAMIA), International Journal of Medical Informatics (IJMI), Artificial Intelligence in Medicine (AIIM), Journal of Biomedical Informatics (JBI) and Methods of Information in Medicine (MIIM). We chose the categories shown in Figure 2 since they are the topics studied in state of the art and, as mentioned before, we intend to demonstrate their importance to the scientific community. The categories are CIG modelling languages, CIG tools, CIG Interaction (Verification of concurrent CIGs), CIG validation and CIG execution engines. Observing the Figure 2, one can see that the categories identified are quite prominent, each including at least four papers from JAMIA, two papers from AIIM and always some other papers from other prominent journals. Some of the themes include at least ten papers each: CIG modelling languages (20), CIG tools (11) and CIG execution engines (15), while other categories show opportunity and need for further research.

Such includes, CIG interaction (verification of concurrent CIGs) and CIG validation and verification. So, through the analysis of the figure, we can conclude that CIG modelling languages and CIG execution engines are the categories with more papers since they represent the main focus of the study of state of the art.

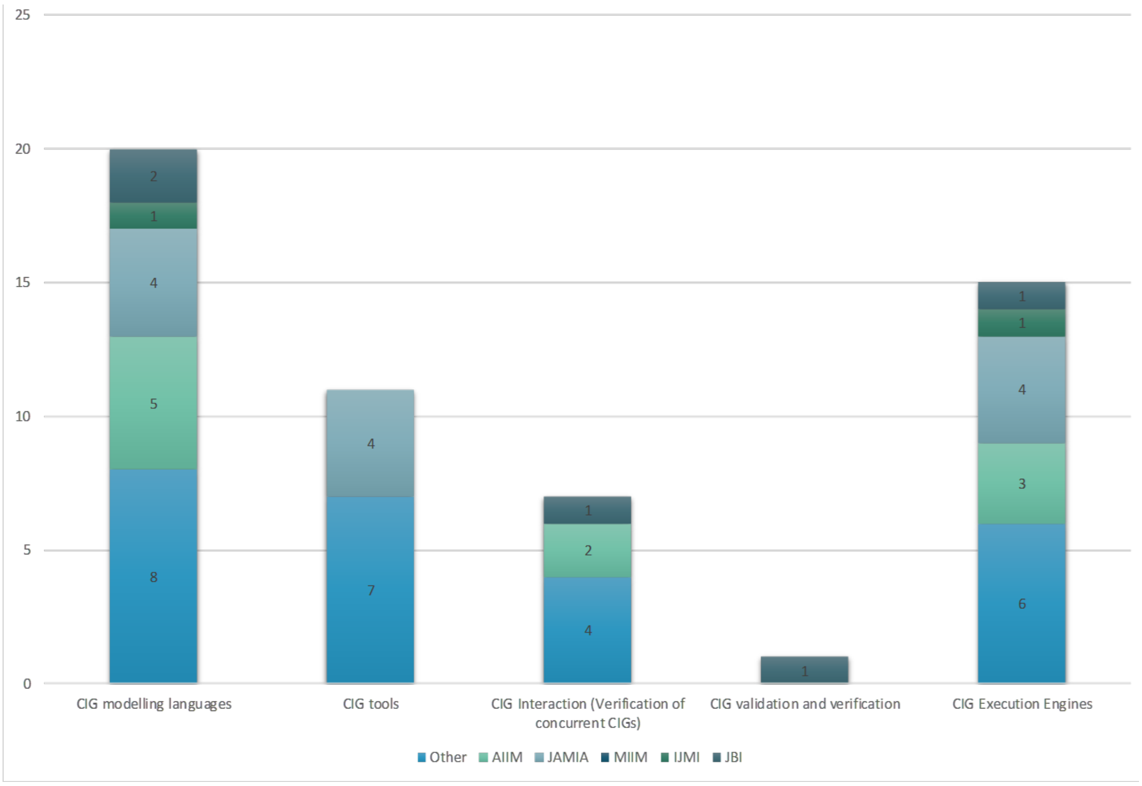


Figure 2: Number of articles published in each category from each journal.

2.1 Existing Systems for CIG Execution

CIGs are representations of CPGs in a structured and machine-readable digital format. They are also representations of CPGs that can be integrated into CDSSs and aim to facilitate the implementation of good recommendations in daily clinical practice, since they are available during the clinical act [36]. The implementation of CPGs in decision support systems can lead to better acceptance and application of these good medical practices because these systems are able to monitor the actions and observations of health professionals [37].

CPGs play an important role in the support to the quality, auditing and research through a set of specifications and requirements used in the health care assessments [38]. Thus, CPGs enable the measurement of the health care in order to evaluate its degree of adequacy. As such, CIGs allow to increase the accuracy, syntactic and semantic correction of CPGs through automated verification tests in order to ensure their integrity. These tests ensure that there are no errors in the protocols since they are based on simulation environments, in which the CPG is applied to different clinical cases, provided by existing patient records.

It can be considered that the first approach to the modelling of CIGs, although rudimentary, was implemented in the HELP system in 1967 [39]. The CIG component of the system consisted of a set of modules, called help sectors, which contained clinical knowledge in the form of logical rules. Since then, the development of CIG approaches has proliferated.

Several CPG description languages exist to create computer-assisted tools for managing protocols that are aimed at representing clinical knowledge. Arguably, the more important are Arden Syntax [20], GLIF [21], Asbru [22], EON [40], PROforma [23] and GLARE [41].

In the following sub-sections, the different approaches to the modelling of CIGs will be presented, paying special attention to the representation model of each one. The selection of the models followed two criteria: consensus of the authors about which are the most relevant and the availability of information in the literature about the approaches.

2.1.1 Arden Syntax

One of the most well-known formalisms in CIG representation and decision support is the Arden Syntax, based on the HELP system. This formalism was developed in 1989 and its development results from a consensus that brought together health institutions and academics, with the aim of creating a model that would allow the sharing of protocols between different institutions, with different information systems [42].

Arden Syntax was recognised as a standard model in 1992 by the American Society for Testing and Materials (ASTM) [43]. Currently, the version of Arden Syntax in development is 3.0 and is distributed by the Health Level Seven (HL7) [44]. This approach focuses on the sharing of simple and independent protocols in the form of modules. It is not an appropriate format for complex protocols, so it is used to represent simple rules.

In Arden Syntax, each CPG is modelled as a Medical Logic Module (MLM) that encodes knowledge for a single decision. Each MLM is an ASCII file that contains components grouped into three categories: *maintenance*, *library*, and *knowledge* [45]. The *maintenance* and *library* components provide descriptive information about the CPG (e.g. title, version, keywords) that are required for its share. The Medical knowledge is stored in the *knowledge* category, through its components: *type*, *data*, *evoke*, *logic* and *action* [45]. The *data* component is used to obtain the values of the concepts referred in the MLM, from the information system of the institution. On the other hand, the *evoke* component specifies the events that trigger the MLM execution. The *logic* compartment contains decision criteria, which can lead to a certain action in the form of production rules, in conjunction with logical operators (e.g. or, and), list operators (e.g. merge, sort), calculation operators (e.g. sum, average), and temporal operators (e.g. before, after, ago). In *action* component, the actions can include sending messages to health care professionals (such as in the case of alert systems) or invoking other MLMs, possible through the call statement.

In terms of language, Arden Syntax is defined in Backus-Naur Form, a notation technique used to describe the syntax of languages used in computing. The MLMs are text-based, where decision criteria are expressed in textual production rules [46].

2.1.2 GLIF

GLIF [47] represents an effort by the Intermed Collaboratory organisation (a collaboration of the Universities of Harvard, Stanford and Columbia) in the development of a shareable representation of CIGs, whose first publication dates back to 1998. It has received influences from other existing approaches, such as Arden Syntax [42], GEODE CM [48], MBTA [49] and EON [40].

This approach was developed to reflect a flowchart of structured and scheduled steps that represent clinical decisions and actions. The first versions of GLIF lacked a formal specification of CPG steps and data mapping for the electronic health record. The main objective of this model is the sharing of CPGs and, so, they are modelled in a way to be perceptible, both by specialists in the clinical domain and by automatic parsers used in different decision support systems.

In its last version, GLIF3 [21] defines five types of steps: *decision steps*, *patient state steps*, *branch steps*, *synchronisation steps* and *action steps*. Each CPG in GLIF consists of a set of points representing each of the five types of steps, linked together in a flowchart. *Decision steps* model decision points in a protocol, directing the workflow of a step to alternative steps. Within a decision, there are two subclasses: *case step* and *choice step*. In *case step* there is a set of logical expressions that direct the workflow to one of the alternative steps. The logical expressions correspond to an excerpt of Arden Syntax, which assumes the designation CPG Expression Language (GEL) [50]. A *choice step* applies in situations where the CPG suggests multiple alternatives, but leaves the choice to an external agent, in this case, the user. *Patient state steps* work as labels that contain attributes used to describe the state of the patient. This type of step can be used as a data entry point in the system. *Branch steps* model a set of concurrent steps, directing the flow of tasks to parallel steps, then synchronized in synchronization steps. Finally, *action steps* model tasks that must be performed. In Asbru, the protocols consist of *plans* and *actions*, and their functionalities are defined by *knowledge roles*. The *preferences* restrict the application of a given plan in order to achieve a given objective.

2.1.3 Asbru

The Asbru formalism [51] allows expressing objectives, which in the context of this model, acquire particular relevance and are designated by intentions. It was developed by Stanford University and the University of Technology of Vienna, and it is particularly advanced in modelling the temporal aspects of CPGs.

Asbru is part of the Asgaard project [52], and it is a formalism for the representation of CPGs as plans, specialising in the representation of patterns and temporal annotations, developing a method of visualising CPGs over a temporal axis. In this model, plans are seen as collections of items. Each task is performed based on problem-solving methods, which present a set of general strategies, regardless of the application domain. The knowledge needed to solve a given task is defined in *knowledge roles*, which describe the knowledge function in the problem-solving method. Asbru defines the following knowledge roles: *preferences*, *plan intentions*, *conditions*, *effects* and *plan body*. The content of a *plan body* is always made up of other plans until a plan can no longer be decomposed. Plans that can not be decomposed are called actions.

One of the key aspects of Asbru is the representation of the goals of a plan in *intentions*. The Definition of *intentions* helps to select the most appropriate plan, and it is crucial in supporting the decision making. An example of such situations is the case in which a health care professional intends to treat hypertension, for which a possible treatment is the administration of adrenergic beta-blockers. However, the doctor may wish to avoid its use and follow another plan. Based on the *intentions*, if the goal of lowering hypertension is not reached, the system is able to make a critical appreciation of the health professional's procedure and advise him. On the other hand, if the treatment outcome is in accordance with the intentions, the critical appreciation will not be generated, and the system will accept the plan followed as correct. The *intentions* are defined as temporal patterns of actions of the health professionals and patient states that must be maintained, achieved or avoided.

In the representation of protocols in Asbru, the temporal annotations are extremely important [53], since they specify four points in time relative to a reference point, which may be a specific or abstract point in time or a transition of states of a plan. These four points are: *earliest starting shift*, *latest starting shift* , *earliest finishing* and *lastest finishing shift*. Two durations can be specified: minimum duration and maximum duration. These references represent temporal constraints for plan execution or condition checking.

Thus, unlike the other approaches such as GLIF and PROforma, the visualisation of CPGs in Asbru does not happen in the form of a flowchart, since the visualisation of time and intentions through a flowchart is difficult and complex. For this purpose, a tool called AsbruView [12] was developed which uses graphics to visualise CPGs. In AsbruView, plans are seen as clues, and the various types of conditions are viewed as traffic signals, according to a topological view. In AsbruView plans can also be seen in the temporal view, which focuses on its time dimension and respective conditions. This view makes use of the temporal references mentioned above.

2.1.4 EON

The EON formalism was one of the CIG models that emphasized the importance of the user understanding the flow of the presented tasks. EON [40] was developed at Stanford University, and it is the precursor of formalisms such as GLIF [50] and SAGE [54], continuing to be developed as a research system. Similarly to GLIF, represents the protocols in the form of flowcharts.

The EON model consists of various components that facilitate the acquisition and execution of CPGs, and it is object-oriented [54]. The EON model is designated Dharma and consists of classes that describe entities of a CPG in the form of temporally structured steps. The Dharma model is not static, which means that the model can be expanded with additional classes that capture new aspects of the CPGs.

In this model, the CPGs manage the patient behaviour, consisting of decisions and actions that can lead to changes in the patient's state over time. In this model, the decisions are made during the consultations of health professionals with patients. Actions, such as writing medical prescriptions or requesting a lab exam, are performed during these consultations. For the representation of time constraints, EON uses an excerpt from Asbru.

The primitive classes in EON are represented in Figure 3 and are *scenarios*, *decisions*, *actions* and *goals*. These classes form the ontology of the model.

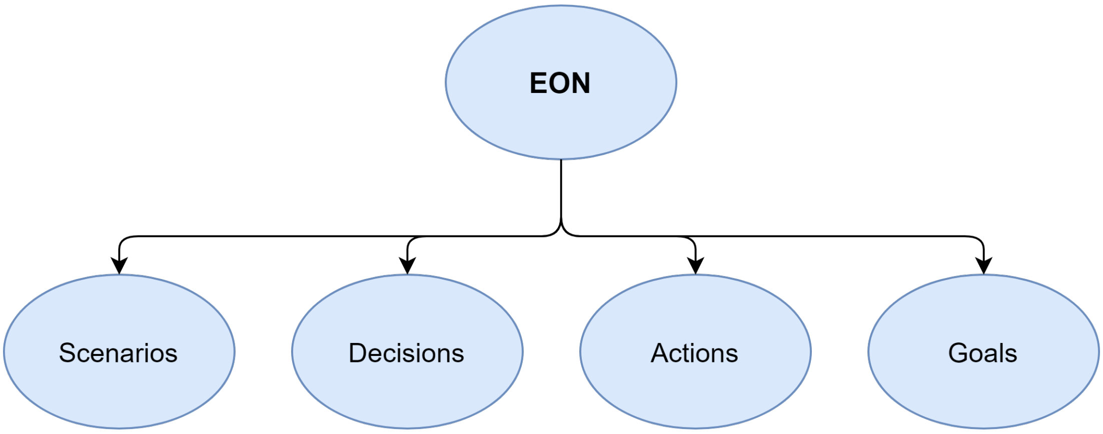


Figure 3: Schematic representation of the main classes of the EON model.

The *actions* are defined as instantaneous acts that lead to changes in the general state, such as collect patient data, show a message to the user or start a new treatment. In EON there is also the concept of *activity*, however, while *actions* are instantaneous, *activities* model processes that take place over time. The *activities* have states that can change over time as a result of *actions*. EON also has *actions* that refer to a set of other actions or a clinical sub-protocol. Examples of this type of *actions* are those that model branching and synchronization steps.

2.1.5 PROforma

In the United Kingdom, the Advanced Computation Laboratory of Cancer Research began the development of the PROforma formalism in 1998 [55] [56]. The purpose of this formalism is the development of more reliable expert systems that can actively assist a patient in health care through decision support and task flow management.

PROforma results from the concatenation of the words proxy (which means permission to act like someone else) and formalize (which means to give definitive form to).

Such as GLIF, PROforma also represents protocols as a flowchart, in which elements are instances of predefined classes. PROforma defines four classes of tasks, depicted in Figure 4.

All tasks derive from a common initial task, root task, which contains attributes that are common to all four derivable tasks. These attributes can be administrative: name (task identifier), caption (task name for print purposes), and description (brief description of the protocol). Other attributes may describe aspects related to the scope of the protocol, such as:

- **Goals**: Define the purpose of the task;
- **Preconditions and postconditions**: They are conditions which must be checked before and after, respectively, of the execution of a task (e.g. risk_level = severe);
- **Trigger Conditions**: These conditions can be temporal events or states of the patient, which, occurring, trigger the performing of a task (e.g. peak_flow<30);
- **Cycles**: Define conditions and restrictions on which a task must be repeated (e.g. cycle(Integer, Interval), cycle(until(State),Interval)).

The *plan* class allows the definition of clinical sub-protocols and determines an ordered sequence of tasks, logical constraints, temporal restrictions on its execution, and the circumstances in which a plan must end or abort, this is explicit in the type of attributes that the *plans* class adds to the already mentioned: *components*, *scheduling constraints*, *temporal constraints*, *abort conditions* and *termination conditions*.

The *components* attribute contains a set of references to tasks that constitute a plan (e.g., history, diagnosis, therapy, follow-up) whose order is defined in *scheduling* and *temporal constraints*. It is possible to define that one task occurs specifically after another, or that a task occurs after a given period of time. This temporal restriction to the performing of tasks is what demarcates PROforma from other approaches like GLIF.

The *decisions* class is represented as a set of candidate solutions. Each candidate solution is associated with a schema with a set of logical expressions that support or refute each result.

An *action* in PROforma is a task whose execution is requested by the PROforma execution engine to an external agent (such as the user, external software, or a hardware component). Typically this type of tasks consists of sending

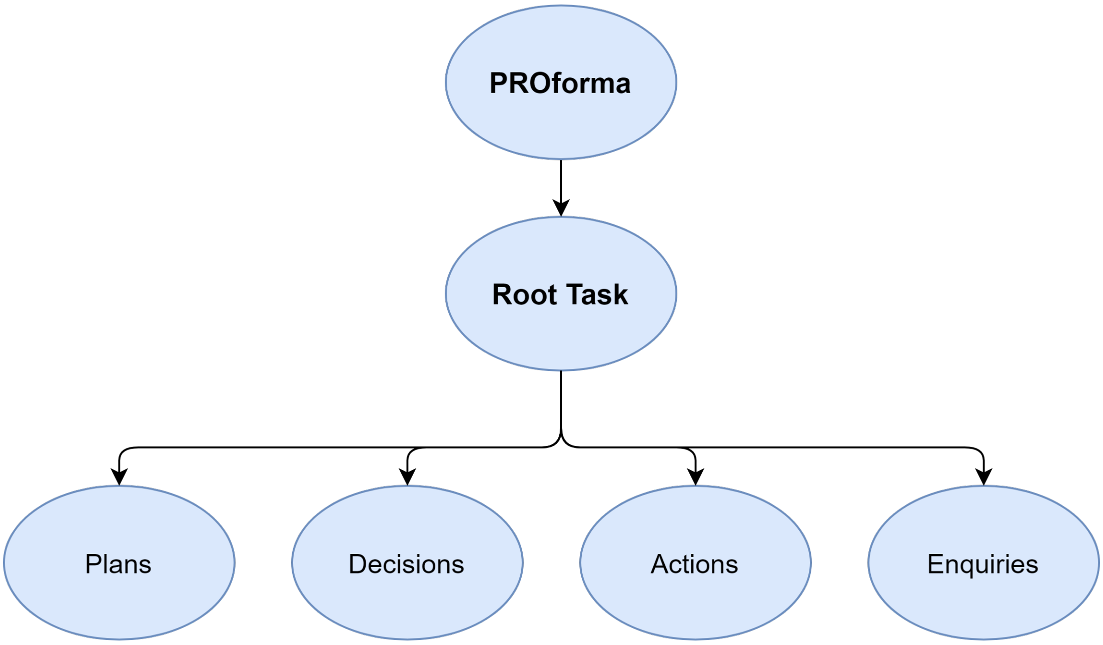


Figure 4: Schematic representation of the main classes of tasks of the PROforma model.

messages or calling an external program through an Application Programming Interface (API). These activities are always atomic and can not be decomposed. In the *actions* class there are the *method* (how the execution should be performed) and *confirmation* attributes (specification on whether the action requires user confirmation).

The *inquiries* class defines tasks for obtaining medical or administrative information. This information can be obtained through questions to the user or from the patient's electronic health record. In this class, we highlight the *data definition* attribute, which defines the way how the data is entered into the knowledge base.

The CPGs in PROforma are stored using Red Representation Language (R^2L), a language of time-oriented representation. To execute the CPGs, it is necessary to convert them to another language, called Logic of $R^2L(L_{(R^2L)}).L_{(R^2L)}$ is a language of predicate-based knowledge representation.

2.1.6 GLARE

GLARE [41] is a project that includes a model for the representation of CPGs and a system for acquiring and executing them. This project was developed by the Department of Computer Science of the University of Piemonte Orientale, Alessandria, Italy.

The representation model does not use a standard representation. Instead, it defines a graph-based framework that displays CPGs, where a clinical action is represented by a node. The main entities of the GLARE model are represented in Figure 5.

It is possible to define atomic actions that represent simple tasks, such as *queries* to obtain external information, work actions that represent medical procedures, decision actions as a set of conditions to select alternatives and conclusions that describe the decision making.

Decision actions are specific types of actions that contain the criteria used to select alternative paths from a CPG. These criteria are represented as triple sets in the form (diagnosis, parameter, result) and, in turn, one parameter is another triple set (data, attribute, value).

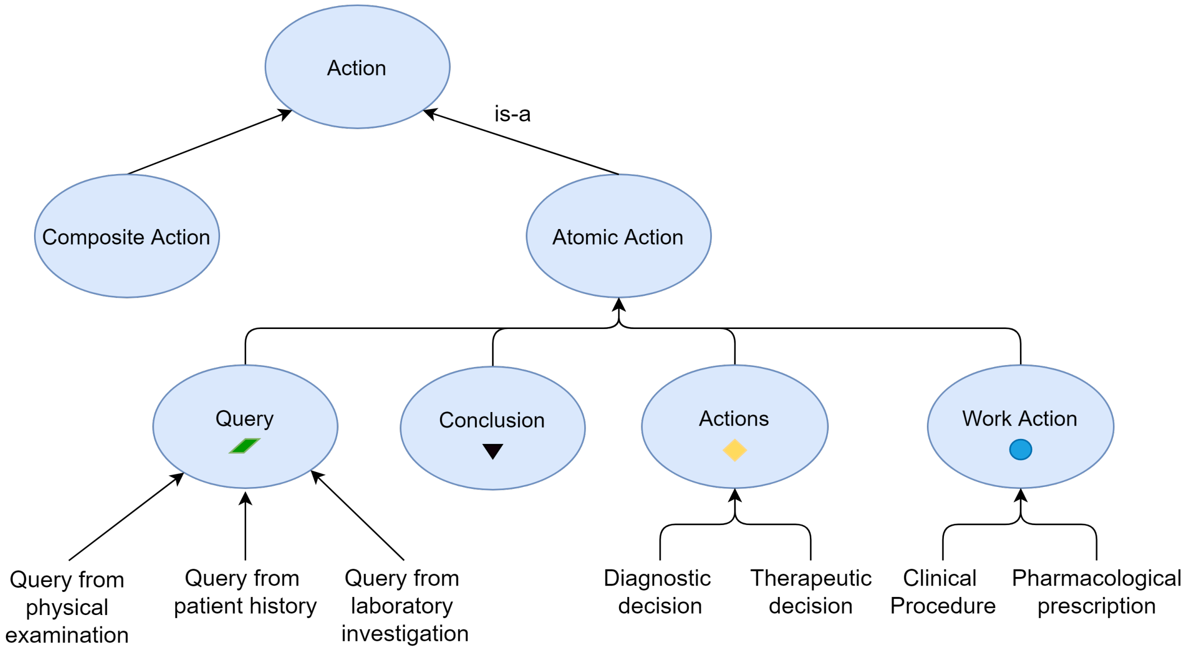


Figure 5: Basic entities defined in the GLARE representation model (taken from [57]).

It is also possible, in GLARE, to define composite actions, which are sets of atomic actions or other composite actions. GLARE is designed to handle different types of time constraints and implements specialized temporal reasoning algorithms.

In the GLARE implementation mechanism [41], there is a distinction between the acquisition phase and the execution phase of the CPGs. GLARE defines three levels of architecture represented in Figure 6, namely System, eXtensible Markup Language (XML), and Database Management System (DBMS).

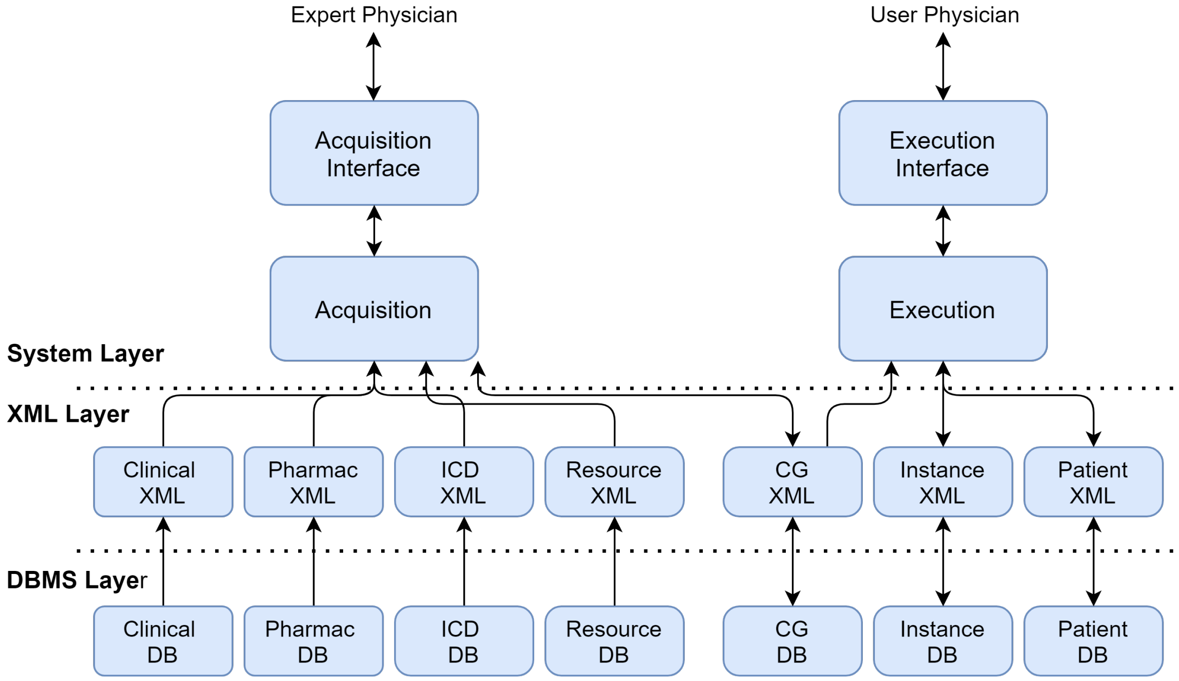


Figure 6: GLARE General Architecture (Figure taken from [57]).

The System level includes the acquisition and execution modules. The XML level is responsible for exchanging data between the System level and the DBMSs level. DBMSs is the lowest level, responsible for establishing a physical connection between higher levels and databases where information for the creation and execution of CPGs is stored. This information includes open instances of CPGs, a repository of protocols and medical records of the patients. GLARE uses ICD-9 as standard terminology.

2.2 Temporal Representation Models of CPGs

The temporal representation and the reasoning in medicine have been explored as research topics since the late 1980s [58].

Most of the clinical tasks require the measurement and recording of diversified and large volumes of patient data, in electronic format. Health professionals who have to make diagnostic or therapeutic decisions based on these data may have difficulties since it can be an overwhelming task.

Thus, it is desirable to provide succinct information, summarised in a temporal context and oriented to clinical data stored in an electronic format that is capable of answering questions about concepts and relationships present in the data.

Even so, one can not put aside another important component in the medical field, and that concerns the different medical tasks, such as decision support for diagnosis, therapy advice, clinical data summarisation and patient monitoring.

One of the most complex clinical activities concerns the application of the CPGs. The scheduling and the temporal management of clinical recommendations is a difficult task in clinical practice since it involves the processing of complex information and a wide variety of complex temporal constraints which affect the starting and finishing times of the recommendations. Therefore, the temporal dimension is an important factor when performing any clinical procedure. So, it is important the ability to represent the time when executing the CPGs resulting from clinical practice.

In most recommendations in CPGs, the procedures should be performed according to a set of time constraints [59]. The first group includes temporal patterns that determine how tasks should be executed, such as:

- **Durations**: constraints that represent how long a task should be performed;
- **Periodicities**: constraints in which a task should be performed from time to time, as a succession of several events;
- **Repetitions**: constraints that define the number of times that a task must be repeated;
- **Waiting Times**: delays in the execution of tasks;
- **Repetition Conditions**: conditions about the state of a patient that determine whether a task should be repeated or not.

The second group consists of temporal constraints that reflect changes occurred, or expected to occur, in the state of a patient.

The temporal management is one of the main concerns in modelling CIGs. Some CIG approaches have been specifically designed to deal with temporal constraints.

The temporal constraints on tasks and conditions about the state of a patient are the most important aspects of the temporal representation of CPGs, so the before-mentioned CIG models will be discussed taking into account these aspects.

All approaches have a form of temporal representation; however, certain models do not develop their constructors, using excerpts from other languages. The most complex, and perhaps most complete structure is Asbru. GLIF [60] and EON [40] adopt an excerpt from Asbru for temporal representation. In order to be compatible with Arden Syntax, GLIF [60] defines a set of common operators to Arden Syntax, such as *before*, *after* and *ago*, which present parallelism with PROforma, allowing to specify the relative order of tasks. PROforma defines a language of expression of restrictions to the execution of plans that allows defining its duration and repetition conditions.

After the study of the different CIG models in section 2.1, it is important to analyse the mentioned projects, as well as to analyse the methodology of temporal representation of the clinical recommendations. Moreover, it is also important to compare these temporal reasoning models and find their limitations. So, in the next sections will be analysed the various approaches regarding the representation of temporal information of CPGs.

2.2.1 GLIF3

GLIF3 [60] addresses both temporal constraints placed on patient state conditions as well as action durations. Asbru [52] also provides a comprehensive representation model for durations. In fact, this CIG model presents the CPGs as structured time-oriented plans for which it is possible to define time annotations, that can be restrictions on the start time and the end time of the tasks (with the beginning as soon as possible and finalization as soon as possible), maximum and minimum durations and cyclical moments (for example, every morning, every day, etc.). This model makes an explicit and extensive treatment of temporal constraints on durations and delays between actions [61], as well as a distinction between parallel actions, sequential actions and cyclical actions.

2.2.2 Asbru

As defined in Asbru, the temporal constraints on the execution of sequential and parallel tasks can be represented in two dimensions, that is, ordering constraints, which can assume the parallel values such as *any order* or *total order*, and *continuation condition* that can assume the values *all completed* or *some completed*.

The combinations of these two dimensions result in five temporal constraints represented by the constructors: *DO-ALL-TOGETHER*, *DO-SOME-TOGETHER*, *DO-ALL-ANY-ORDER*, *DO-SOME-ANY-ORDER*, and *DO-ALL-SEQUENTIALLY* [62].

As already mentioned, Asbru also provides a third category of temporal constraints, used to define periodicities. All of these constraints are represented within the main plan of the protocol. The same approach is used by EON and then adopted by GLIF through an excerpt from Asbru for temporal representation.

2.2.3 GLARE

In GLARE there was an evolution regarding temporal representation [63]. In this model, more complete constructors were introduced for the representation of periodicities and repetition schemes. This formalism was later expanded by

Anselma et al. in [59]. The new version provides an improved formalism to express periodicities, with the possibility of defining the delays between the cycles of the periodic event, making it possible to define more complex periodicity patterns. For example, each cycle of a periodic event may have an associated periodicity.

Another interesting development is the mapping of temporal patterns to a high level of abstraction by Anselma et al. in [59], for a Simple Temporal Problem (STP) [64], a reasoning framework in which a representation in the graph form of the CPG is provided, resulting from the calculation of the time limitations related to each task. Although it is one of the most used temporal reasoning techniques in AI, the STP does not allow to represent complex temporal patterns, since the STP-tree structure produced is not suitable for events that repeat over time [64]. This aspect reflects the dominant view that the pure AI approaches [9], mainly derived from logic, are faced with obstacles when applied to the medical domain.

2.2.4 PROforma

In PROforma [55], the CPGs are modelled as plans and each plan can define constraints on task execution, as well as task duration and task delays. Besides, the temporal constructions can also be used to specify the preconditions for actions.

2.2.5 EON

EON [40] uses temporal expressions to allow planning steps on CPGs and deals with duration constraints on activities. On the other hand, through the incorporation of the RESUME system, a powerful approach is provided to deal with temporal abstraction. In EON and Arden Syntax there is the representation of delays between the event that activates an MLM and between MLMs [65].

In the next section, a comparison of the temporal representation models of CPGs is presented.

2.2.6 Discussion and Analysis of Temporal Representation Models

The comparison of CIG models based on temporal constrains is explored in depth in section 3.1. Table 1 represents the comparison of CIG models based on temporal constraints on task execution and temporal constraints on the state of a patient. In general, all approaches present a reasonable number of temporal constructs, associated with temporal constraints on task execution.

Table 1: Assessment of CIG models. The symbol ✓indicates the model fully represents the temporal constraint and the ✗ indicates the model does not represent it or has limitations regarding it.

CIG Model	**Temporal restrictions about the execution of tasks**					**Temporal restrictions about the state of a patient**
	Durations	Repetitions	Periodicities	Waiting Times	Repetition Conditions	
Arden Syntax [20]	✓	✗	✗	✓	✗	✗
GLIF3 [60]	✓	✗	✗	✗	✗	✓
Asbru [52]	✓	✓	✓	✓	✗	✗
EON [40]	✓	✗	✗	✓	✓	✗
PROforma [56]	✓	✓	✗	✓	✓	✗
GLARE [59]	✓	✓	✓	✓	✓	✗

Regarding the temporal pattern *duration*, all approaches present this constructor, but Asbru has a complete view, in that it allows defining this constructor, through a set of intervals.

The temporal pattern *repetition* is not represented by EON, GLIF3 and Arden Syntax. The remaining approaches define this type of constructor in the temporal constraints of CPGs. EON, PROforma and Luca Anselma represent repetition conditions, and the other approaches do not define this constructor.

About the representation of periodic events, Abru, PROforma, GLARE and the approach of Anselma et al. [59] represent this temporal pattern of CPGs, with greater emphasis being placed on GLARE since it is possible to define this temporal constructor through a detailed set of intervals.

Regarding the representation of temporal constraints on the state of a patient, only GLIF3 focuses on this point, so the remaining approaches present a gap in this aspect.

In conclusion, it is possible to admit that there is a lack of an approach that includes all these aspects of time constraints, which proves the need to create a comprehensive model capable of accommodating the temporal patterns previously identified. To address the mentioned limitations, we developed a model that is described in the article of section 3.1.

2.3 CIG Tools

Several projects were proposed in literature for helping users and developers in creating a effective CIGs that cope their definitions, standards and constraints. These tools are aimed at helping users create, either manually or semi-automatically categories, partonomies, taxonomies, and other organisation levels of CIGs [66]. The most prominent CIG editors and managers are:

- *Protégé Desktop*;
- *SAGE Workbench*;
- *Tallis*;
- *GEM Cutter*;
- *Asbru View*.

In the next sections, the before-mentioned projects will be explained in more detail.

2.3.1 Protégé Desktop

Protégé [67] is an open-source tool produced at Stanford Medical Informatics [68], which provides an environment for ontology development and knowledge acquisition. It presents a graphical tool that provides an extensible architecture for creating customised knowledge-based tools and constructing large electronic knowledge bases. *Protégé* provides two forms of modelling ontologies:

- **Protégé-Frames editor**: its model is compatible with the protocol of Open Knowledge Base Connectivity (OKBC). Accordingly, all entities (i.e., instances, classes, slots, facets, and constraints) are frames. Instances represent domain objects. Classes can represent both named collections of instances or abstract conceptual entities in the domain (e.g., the notion of a medicine ingredient). Slots represent binary relations that describe classes' properties (e.g., drug implications). Facets describe properties of slots (e.g., the data type of a slot's value). Constraints specify extra relationships that exist between instances;

- **Protégé-OWL editor**: permits to build ontologies in OWL.

Protégé supports the formalisation of a domain ontology, the design of customised knowledge acquisition forms, and entering domain information that can be adjusted to allow conceptual modelling with new and emerging Semantic Web languages. *Protégé* supports the analysis of domain models at a conceptual level without knowing the language syntax. Thus, it permits to focus on the concepts and relationships in the domain [69].

It presents a platform that can be extended with graphical widgets for tables, diagrams, and animation components to access other knowledge-based systems embedded applications. It also allows other applications to use its libraries for accessing and displaying their knowledge bases. Moreover, it can be extended using plug-in architecture and a Java-based API for building knowledge-based tools and applications. *Protégé* is used to author CPGs in various models. Different components of the modelling process can be performed using predefined graphical symbols, organised in a diagram and linked by graphs. The data entry points can be filled using forms [70].

Protégé allows the export of the ontologies into various formats, including RDF(S), OWL, and XML Schema [71]. The Figure Protégé Desktop application interface is shown in Figure 7.

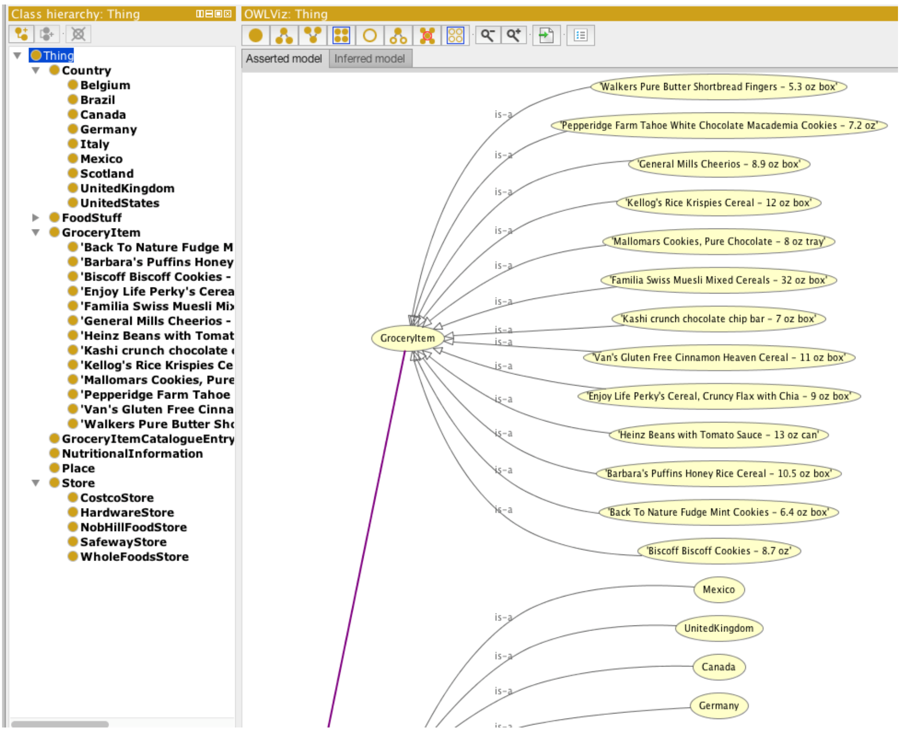

Class hierarchy: Thing
OWLViz: Thing
Asserted model
Inferred model
Thing
Country
Belgium
Brazil
Canada
Germany
Italy
Mexico
Scotland
UnitedKingdom
UnitedStates
FoodStuff
GroceryItem
'Back To Nature Fudge M
'Barbara's Puffins Honey
'Biscoff Biscoff Cookies -
'Enjoy Life Perky's Cerea
'Familia Swiss Muesli Mix
'General Mills Cheerios -
'Heinz Beans with Tomat
'Kashi crunch chocolate c
'Kellog's Rice Krispies Ce
'Mallomars Cookies, Pure
'Pepperidge Farm Tahoe
'Van's Gluten Free Cinna
'Walkers Pure Butter Sho
GroceryItemCatalogueEntry
NutritionalInformation
Place
Store
CostcoStore
HardwareStore
NobHillFoodStore
SafewayStore
WholeFoodsStore
GroceryItem
is-a
'Walkers Pure Butter Shortbread Fingers – 5.3 oz box'
'Pepperidge Farm Tahoe White Chocolate Macademia Cookies – 7.2 oz'
'General Mills Cheerios – 8.9 oz box'
'Kellog's Rice Krispies Cereal – 12 oz box'
'Mallomars Cookies, Pure Chocolate – 8 oz tray'
'Familia Swiss Muesli Mixed Cereals – 32 oz box'
'Kashi crunch chocolate chip bar – 7 oz box'
'Van's Gluten Free Cinnamon Heaven Cereal – 11 oz box'
'Enjoy Life Perky's Cereal, Cruncy Flax with Chia – 9 oz box'
'Heinz Beans with Tomato Sauce – 13 oz can'
'Barbara's Puffins Honey Rice Cereal – 10.5 oz box'
'Back To Nature Fudge Mint Cookies – 6.4 oz box'
'Biscoff Biscoff Cookies – 8.7 oz'
Mexico
UnitedKingdom
Canada
Germany

Figure 7: The interface of Protégé Desktop that depicts a CPG workflow (extracted from OWLViz Git Project[1]).

[1] https://github.com/protegeproject/owlviz/

2.3.2 SAGE Workbench

The *SAGE* Workbench is a complete, self-contained environment that uses *SAGE* CPG model. This model encodes CPG knowledge needed to provide situation-specific decision support and use standardised interoperability components. *SAGE* Workbench provides a tool for developing knowledge authoring models based on Protégé. Also, SAGE defines the knowledge deployment process, and knowledge execution architecture [72]. *SAGE* CPG Workbench includes a set of features that allows creating, viewing, editing, and validating *SAGE* CPGs according to *SAGE* CPG Model format. The project has several requirements for the CPG workbench. It should be a tool that:

- Supports encoding process;
- Link terminology services that can be used during the CPGs encoding;
- Allows debugging and validating the CPGs;
- Provides a document-oriented view of the CPG knowledge units for clinicians, and knowledge experts can easily view and interpret its content.

SAGE Workbench additionally incorporates a terminology plug-in (the *SAGE* DTS tab) which accesses via the Internet the Apelon Distributed Terminology System (DTS) terminology service (developed by Apelon, Inc., USA). This plug-in allows view standard and *SAGE*-based terminologies, do concept queries, and view complex logical concept expressions. The Apelon DTS utilises client-server technology, which requires user authentication into the DTS server for accessing its functionalities. It is important to emphasise that it does not require access to Apelon's terminology service to use the workbench or display the CPGs.

One of its main features is to integrate CPG-based decision support with the care process's workflow. Also, *SAGE* is recognised as a prominent model for CDSS with a extensive knowledge base coverage. *SAGE* includes a knowledge authoring tool based on *Protégé*.

Therefore, SAGE can be robust, and a comprehensive knowledge representation model for clinicians [73], which allows representing several clinical concepts.

SAGE Workbench was developed for functioning in MS Windows computers, and its interfaces are organised in several tabs, as shown in Figure 8.

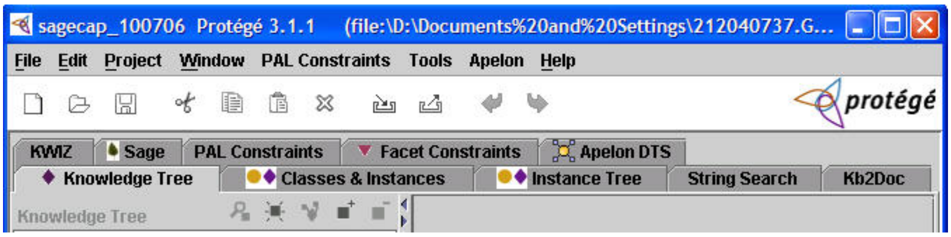


Figure 8: The different tabs provided by Protégé Desktop platform using SAGE Workbench Plug-in (extracted from Sage Website[2].

[2] http://sage.wherever.org/encoding/encoding_tools.html

KnowledgeTree Tab allows the exploration of frames directly and indirectly referenced from a selected instance in a tree structure. It allows to browse and edit, in a single window, all the frames reachable from the top-level instance (usually the top-level CPG instance).

Facet Constraints Tab supports identifying and rectifying all instances in a knowledge base, or instances of selected classes, which have slots with values violating the slot's constraints. The Facet Constraints Tab allows to easily navigate through an ontology (it only requires one click) and to identify instances that had constraint violating facets in the ontology.

Protégé Axiom Language (PAL) is part of first-order logic that allows expressing integrity constraints about a knowledge base. The PAL Constraints Tab is a front-end for this constraint system. This Tab allows creating, browsing, and modifying constraints in the knowledge base, and evaluating constraints (either as a group or individually).

Apelon terminology Tab and plug-in (i.e. and integration between *Apelon* software and *Protégé*) work with Apelon DTS 3.0 server across the Internet. This plug-in allows a search of terms from several terminologies and creating a reference in the *Protégé* CPG Workbench to a term in standard terminology.

Kwiz Tab allows customising high-level views of the knowledge base, constrained navigation, reuse of existing knowledge bases, context-sensitive search and help.

SAGE Tab provides a self-contained testing environment within *Protégé* for encoded CPGs.After verifying the knowledge base, it is possible to select data from a test case and run a CPG by simulating the triggering events. After evaluating possible immunisations that can be triggered, it requests information on immunisation consent and serious diseases that may render improper immunisation. Once answers to the questions are submitted, the *SAGE* Execution Engine will generate its final recommendations for the immunisations that should be given.

Kb-to-doc Document-generation Tab allows generating a document-oriented view of the contents of the encoded CPG.

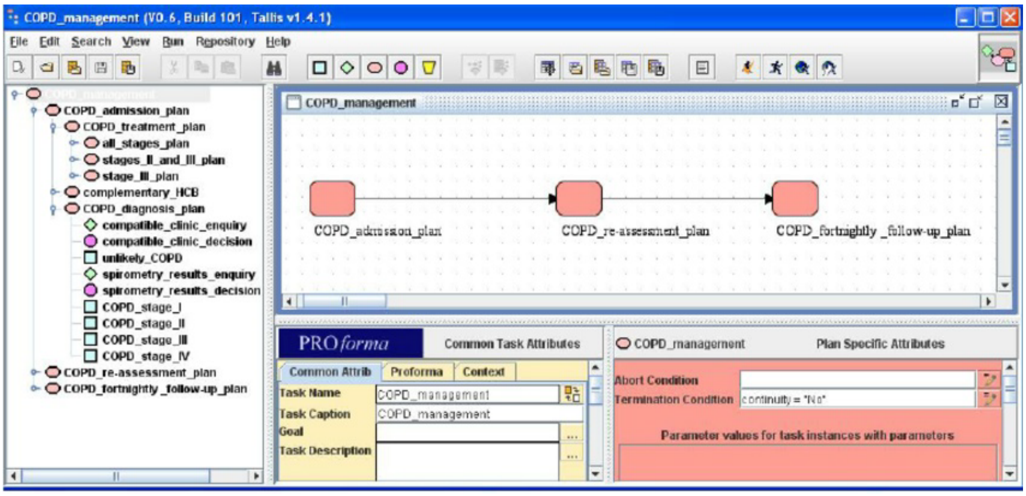
2.3.3 Tallis

Tallis is a recent version of PROforma-based models implementation that provides an execution tool. It was produced by the Cancer Research UK [74]. *Tallis* consists of a set of applications, which are as follows:

- Composer - provides a tool for authoring PROforma CPGs in a graphical editor. Composer incorporates a test platform for simulating interaction with PROforma CPGs, default settings for setting up applications, and a repository for saving applications, so-called OpenClinical repository;
- Tester - an application that allows testing PROforma CPGs. Although tester application comes bundled with Composer, it can be run as a standalone application;
- Web Enactment - a web server application for enacting CPGs on the web, used in OpenClinical.

These multi-platform applications can be integrated with other components, including third-party applications. It requires the *Tallis* engine and core plugins [75].

PROforma allows representing the activities that need to be performed by an agent. These agents can be modulated with objectives to be achieved in a particular situation, and under various practical constraints (e.g. timing,



Task Name: COPD_management
Task Caption: COPD_management
Goal:
Task Description:
Abort Condition:
Termination Condition: continuity = "No"

Figure 9: Tallis Composer Interface (extracted from [77])

resources or information constraints). It combines features of a specification language as developed in software engineering, and a knowledge representation language developed in AI [76]. The user interface for the *Tallis* composer is shown in Figure 9.

Process-descriptions are displayed in *Tallis* in two views, a network and tree view:

- The tree view displays the process-descriptions hierarchical structure;
- The network view shows the order of tasks according to scheduling constraints.

A process-description hierarchy is based on plans: each plan defines a new level in the hierarchy.

- Plan contents can be visualised at a glimpse in the tree view;
- The network view permits better visualisation of the details of a plan at a time.

Tasks' properties can be accessed or managed in the Task Properties window by clicking on the tree's task or the network area.

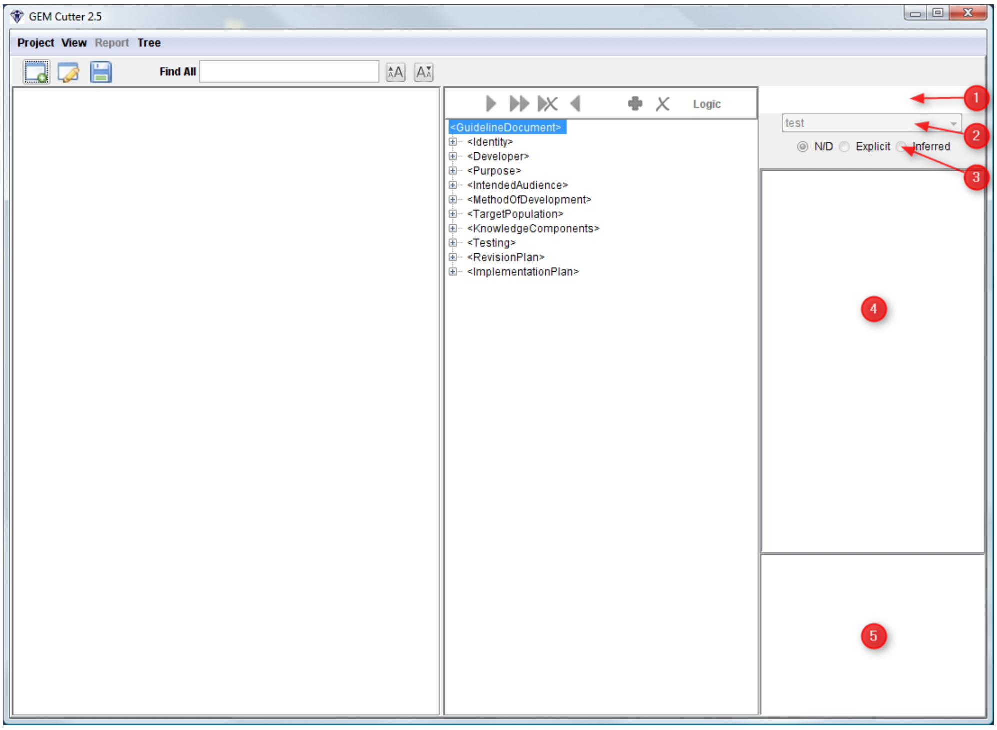
More details can be found in the user-guide found on their website [78].

2.3.4 GEM Cutter

GEM Cutter is an XML editor, whose idea is to facilitate the conversion of CPG text versions into markup language. It supports the conversion of a CPG document into the *GEM* format, and publication in a cross-platform manner [79].

The *GEM* model is an XML-based document that can store and organise the different information contained in CPGs. Its main idea is to facilitate the corresponding CPG text versions into a format that computers can process.

GEM is used throughout the entire CPG life-cycle to model information about CPG development, dissemination, implementation, and maintenance. It allows representing information in high and low levels of abstraction. By using XML standard for information representation, it facilitates computer processing of the CPG information [81].



1 - Element Name; 2 - Action Type; 3 - Element Source; 4 - Element Text; 5 - Element Definitions.

Figure 10: GEM Cutter Interface (extracted from [80]).

The user interface for the *GEM Cutter* is shown in Figure 10. The *GEM Cutter* main window contains three panels, a menu bar, and button bar. The left panel provides the CPG details, while the middle panel contains the CPG tree view in the *GEM* hierarchy. The right panel display five areas as illustrated in the figure:

1. **Element Name**: Displays the name of the selected instance of the tree view;

2. **Action Type**: Drop-down list of action types;

3. **Element Source**: Presents the source of the selected element in the tree view. The source value is determined by how the data was supplied to the GEM document;

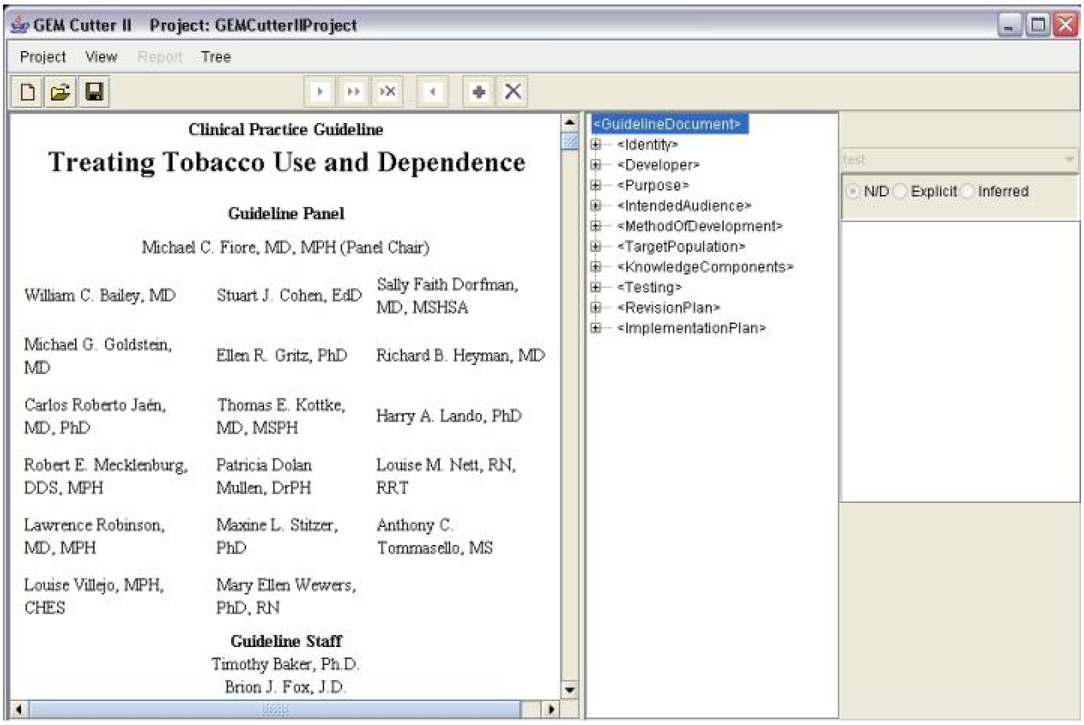
4. **Element Text**: Contains the entire text of the currently selected element in the tree view. Data can be filled directly or edited in this window;

5. **Element Definitions**: Contains the definition of the currently selected element in the tree view.

Figure 11 shows the GEM Cutter interface after opening a CPG. Details about GEM Cutter application are provided in the user-guide found on their website [82].

2.3.5 Asbru View

Asbru View provides a tool that permits physicians to overview of a plan hierarchy. *Asbru View* is based on visual metaphors to facilitate the interpretation and overview of underlying concepts. In other words, *Asbru View* is a



Clinical Practice Guideline

Treating Tobacco Use and Dependence

Guideline Panel

Michael C. Fiore, MD, MPH (Panel Chair)

William C. Bailey, MD	Stuart J. Cohen, EdD	Sally Faith Dorfman, MD, MSHSA
Michael G. Goldstein, MD	Ellen R. Gritz, PhD	Richard B. Heyman, MD
Carlos Roberto Jaén, MD, PhD	Thomas E. Kottke, MD, MSPH	Harry A. Lando, PhD
Robert E. Mecklenburg, DDS, MPH	Patricia Dolan Mullen, DrPH	Louise M. Nett, RN, RRT
Lawrence Robinson, MD, MPH	Maxine L. Stitzer, PhD	Anthony C. Tommasello, MS
Louise Villejo, MPH, CHES	Mary Ellen Wewers, PhD, RN	

Guideline Staff

Timothy Baker, Ph.D.

Brion J. Fox, J.D.

Figure 11: Treating Tobacco Use and Dependence CPG Example in GEM Cutter (extracted from [80]).

graphical interface for viewing, creating and modifying *Asbru* plans. It displays different views of the different aspects of plans [83].

Asbru View consists of two main views: Topological View (TopoView) and Temporal View (TempView):

- The Topological View essentially presents the relationships between plans, without a precise time scale. The basic metaphor in this view is the running track;

- The Temporal View focuses on the temporal aspects of plans and conditions. In addition to the topological information, the Temporal View allows visualising the details of the temporal extensions of Plan.

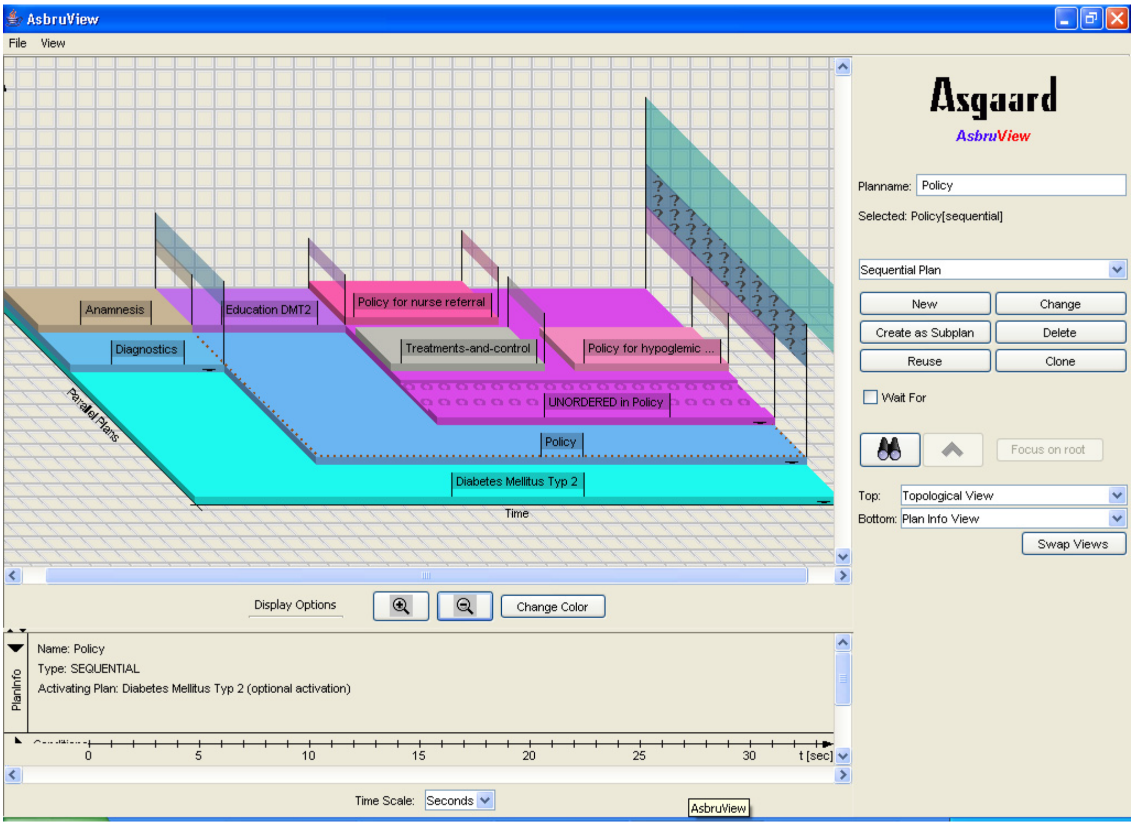
Figure 12 depicts the interface of Asbru View. *Asbru View* main window consists of four panels, a Menu Bar, a Control Panel, an Upper view pane and a Lower view pane. Each of these panels consists of a set of controls explained briefly in the following list:

- **Menu Bar**: Provides global commands such as New, Quit, and others;

- **Control Panel**: This panel provides functionalities that allow managing plans, to focus on plans and the different views of the right side of the control panel;

- **Upper view pane**: Permits the plan's visualisation in an upper view. The screen-shot shows the Topology View;

- **Lower view pane**: Lower view. Can be hidden.



Planname: Policy

Selected: Policy[sequential]

Sequential Plan

□ Wait For

Top: Topological View

Bottom: Plan Info View

Name: Policy

Type: SEQUENTIAL

Activating Plan: Diabetes Mellitus Typ 2 (optional activation)

Time Scale: Seconds

Figure 12: Asbru View Interface (extracted from [84]).

2.3.6 Discussion and Analysis of Computer-Interpretable Guideline Tools

In the next section, a discussion comparing each application's main aspects and limitations will be performed. For this purpose, a set of comparative features was selected to analyse and evaluate the different projects. It is important to note that most features are related to the user experience [85]. This set of features are as follows:

1. **Graphical CPGs View**: Graphical representation (tree, node-link, network diagrams) of parts of or a full CIG workflow. The organisation of the content benefits the understanding the workflow, identify relevant points of the CPG, and manipulate knowledge elements;

2. **Organization**: It is related with the simplicity of using the tool, determined by its structure and how its functionalities are made available, whether they are placed correctly, under the right menu. This feature allows a better understanding of the application structure, optimising its use;

3. **Simplicity**: I mean the ease of access to the tool's functionalities. If it is complex and challenging to use a feature, it leads to confusion, which in turn can difficult the user adoption of the application;

4. **Automation**: While managing new instances, it only should require relevant knowledge elements to be filled, while the system should automatically process the other entry points;

5. **Drag-and-Drop**: The ability to drag-and-drop instances of a CPG in a GraphicalView and filter the workflow of the CIG with the help of graphical type links;

6. **Web/Local Repository**: The opportunity to save or retrieve CIGs either locally or remotely (i.e., in a cloud repository).

Table 2, we compare the platforms studied herein using the aforementioned comparative features. It is important to state that this comparison does not include plug-ins applied to these platforms to add new features.

Table 2: Comparison of the different CIGs Tools. The symbol ✓indicates the model fully represents the feature and the ✗ indicates the model does not represent it or has limitations regarding it

Feature/Platform	*Protégé Desktop*	*SAGE Workbench*	*Tallis*	*GEM Cutter*	*Asbru View*
Graphical CPGs View	✓	✓	✓	✗	✓
Organization	✓	✓	✓	✓	✓
Simplicity	✗	✗	✓	✓	✓
Automation	✗	✗	✗	✗	✗
Drag-and-Drop	✗	✗	✗	✗	✗
Local Repository	✓	✓	✓	✓	✓
Web Repository	✓	✓	✗	✗	✗

Although all the tools display the main features (the modelling of ontologies), they present past applications' characteristics, concentrating only on the proper functioning and less on appearance or ease of management.

One of the features that studied platforms lack is the ability to deploy Automation data, which shows opportunities to be explored by new projects. This feature's main idea is to fill relevant data entry points, leaving the other information be automatically processed by the system. Additionally, all data managed by the platform should be displayed in a simple and organised graphical format (a good layout promotes the understanding of managed data, especially for users with less informatics knowledge) to facilitate the process of fill the data entries.

Although the *Protégé Desktop* and *SAGE Workshop* provide many useful features, the number of menus that they display is significant, making it difficult for the user understanding the different functionalities. The Drag-and-Drop is a crucial feature that enables the administration of CIGs in a visual and straightforward layout and user-friendly input form.

Another relevant characteristic is the opportunity of importing or exporting CIGs saved locally or in a cloud. The ability to store data in clouds is a must-have feature since it allows access all the data anywhere, anytime.

2.4 Combining CPGs - Multimorbidity

During clinical practice, the health professional faces conflicting situations that may impair the patient's clinical condition. The presence of different clinical conditions results in applying multiple-specific clinical CPGs; whence multiple interactions must be considered. It includes complex multiple drug regimens (polypharmacy) with the potential for harmful combinations of drugs, increased treatment complexity, the burden of disease and cost of treatment [29].

MCDA can help reasoning about these interactions by providing methods to compare and evaluate the different decision alternatives, the capability of assessing and integrating multiple criteria, the possibility of structure and assessment of a complex problem, the possibility of leading with incomplete and uncertain information and helps stakeholders summarise complex value trade-offs consistently and transparently helping to do fairer decision-making. The MCDA approach follows a set of fundamental steps regarding its operation. These steps concern how to conduct the MCDA process and they include the following [86]:

1. **Defining the decision problem:** It involves identifying and understanding the decision problem and defining the goals that will guide the decision process. It is also essential to identify the stakeholders, the decision alternatives under consideration and the output required. The stakeholders can act on behalf of others or can be the parties involved in the decision-making process. Generally, stakeholders include nurses, physicians, patients, and others. The types of decision problems can include a ranking/categorising the risk/benefit of the alternatives and understanding the value of alternatives;

2. **Selecting and structuring criteria:** The next step is to identify the criteria by which the decision alternatives must be evaluated. The selection of criteria must be consistent with the decision; the criteria should be independent of each other, represented on the same scale and should not be related to alternatives;

3. **Weighting criteria:** It involves eliciting stakeholder preferences amongst criteria. The weights reflect trade-offs between criteria and clarify the positions of each stakeholder. These weights are used to combine individual criteria scores into a measure of ”total value”. This process promotes transparency of the values considered by enabling the consultation of large groups of stakeholders, mainly through surveys;

4. **Measuring performance:** In this step, it is crucial to determine the performance of alternatives within each criterion. The performance of alternatives can be drawn in a matrix, called the performance matrix, which reminds decision-maker deliberations;

5. **Scoring alternatives:** Before analysing the alternatives' performance, it is essential to capture the stakeholder preferences on alternatives. This can be accomplished by using functions or rules that convert the performance measurements in scores. These scores can be converted in two ways: in units (e.g. 0 to 10 scale) or intervals to incorporate preferences in a scale (e.g. 0-10 or 20-30). There are two methods of elicitation commonly used, namely compositional and decompositional. Compositional techniques generate separate scores and weights for each criterion and are combined in an overall score, whereas in decompositional methods, the weights and scores are derived from the overall values, using regression-based techniques;

6. **Data aggregation method:** This step can have different ways of being represented: a product, an average or a function. The result of applying this method will separate the best alternative from all the others that have been selected. The idea is to produce total values, using alternative scores on the criteria and the weights for the criteria. These total values allow ranking alternatives to identify the degree to which one decision alternative is preferred over another;

7. **Dealing with uncertainty:** The decision alternatives, the underlying criteria weights and performance scores are subject to uncertainty. This may affect both the design and evidence feeding into the assessment. Performing uncertainty analysis can help to understand the robustness of the MCDA results;

8. **Reporting results:** The last step comprises presenting the results. This can be accomplished by presenting them in a graphical or tabular view. These views allow giving the ranks of alternatives to facilitate the understanding of the importance of one decision alternative over another. The decision-makers can also be presented with the MCDA model to explore the results for other case scenarios.

For determining weights, three types of approaches are used in MCDA [87]. These approaches can be classified into Value Measurement Models, Outranking models, and Reference-level Models as depicted in Figure 13.

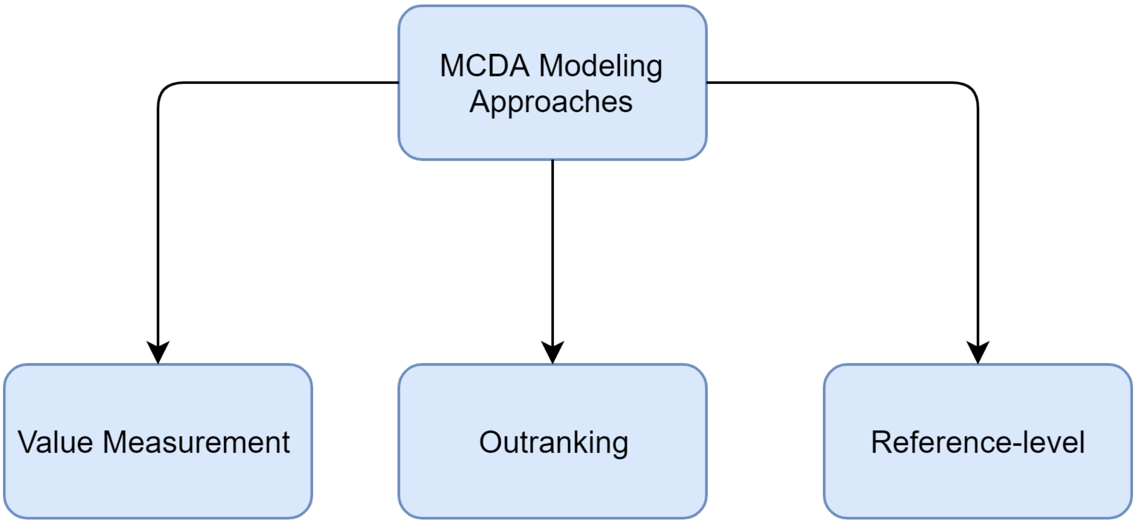


Figure 13: MCDA Modelling Approaches

The details of each approach are as follows:

- **Value Measurement Models:** This method constructs and compares numerical scores, to identify the degree in which one decision alternative is preferred on another. It uses additive models that multiply each alternative's score by the weight relative to a certain criterion. Finally, the weighted scores related to each criterion are summed to obtain each alternative's total score;
- **Outranking Models:** In this approach it is performed a paired comparison of alternatives in each selected criteria. For this purpose, the different criteria are combined to obtain a measure that allows supporting the selected alternative. The selected alternative is based on a rank of alternative solutions prior performed, and typically the alternative with the highest score. One characteristic of this method is that it allows, respecting certain conditions; two alternatives to be classified with incomparable. This incompatibility may be associated with the lack of information for these alternatives when assessing the different alternatives;
- **Reference-level Models:** It requires searching for the best alternative, depending on minimum threshold levels, which are prior defined for each criterion. This approach is mostly based on linear programming techniques.

Typically, Value Measurement Models are the preferred modelling approach, while other techniques are rarely used in healthcare [88]. Although these models are used and applied for different purposes, most of them follow the same prior identified steps. Moreover, MCDA is not a strict process as this is an iterative method that allows performing all the steps in a different sequence. In the context of this doctoral project, MCDA provides specific methods to analyse recommendation conflicts and elicit stakeholder preferences on best decision alternatives. This approach allows constructing and comparing numerical scores (overall value) to identify the degree to which one decision alternative is preferred over another and produce a rank of alternatives. When eliciting stakeholder weights on criteria and capture scores on alternatives, it means getting patient's preferences over clinical recommendations and the clinician's priorities over goals. This process should result from a discussion between the patient and the physician and should be taken into account when applying a patient's treatment plan. In the next section, we will analyse projects for Benefit-risk Assessment (BRA) and Shared Decision-Making (SDM).

2.4.1 Shared Decision Making and Benefit-Risk Assessment Models

The European Medicines Agency (EMA) Benefit-Risk Methodology Project [89] was developed BRA of medical drugs. The authors investigated the applicability of the current MCDA tools and processes for BRA and performed tests of different methods in five regulatory agencies of European medicine.

The IMI PROJECT [90] was created to assess the safety, efficacy and performance of products. They evaluated drugs such as efalizumab, natalizumab, rimonabant, rosiglitazone, and telithromycin with MCDA. They concluded that MCDA provides structured stepwise instructions with the capability to assess and integrate multiple benefit and risk criteria and compare different alternatives. According to the IMI PROTECT, there are attractive features that promote the usage of MCDA for BRA. These features pertain to the fact that MCDA is the only approach that can formally deal with multiple objectives simultaneously, and provide several software packages to perform the analysis Colo-rectal Cancer Screening.

In [91], the authors used the Analytic Hierarchy Process (AHP), an MCDA technique, to extract the decision priorities of people at average risk in Colo-rectal cancer screening. Based on American CPG statements, they identified six criteria: the ability to prevent cancer, avoidance of side effects, minimising false positives, and logistical complexity, further divided into three sub-criteria: frequency of testing, preparation required, and method of the testing procedure.

In [92], the authors translate existing conceptual descriptions into a three-step model for routine clinical practice. They proposed an SDM model based on three key steps: *choice talk*, *option talk* and *decision talk*. In *choice talk*, the set of available options is presented. In *option talk*, more details about the options are given. Finally, in *decision talk*, it is necessary to decide the best option considering the preferences. Throughout the process, the clinician helps patients to deliberate and express their preferences.

Our approach is inspired by the above-mentioned MCDA applications to health care, namely for BRA and SDM, as analysed in article of section 3.4. Also, to mitigate possible interactions between CPGs we utilise an argumentation framework called ASPIC+G, as analised in article of section 3.5.

2.4.2 Discussion of the MCDA Approach

The ability to evaluate and integrate various criteria and compare and evaluate different decision alternatives are some of the advantages of the MCDA approach. Another advantage is dealing with incomplete or uncertain information [93], helping the various stakeholders make fairer and well informed clinical decisions.

Despite these advantages, there are some limitations in using this method in a decision making process. Although many MCDA methods and models exist to evaluate the best decision, none can be considered the "best"appropriate for all situations. Thus objective guidelines that help choose an appropriate MCDA method are rare [94]. Another limitation concerns that MCDA methods are suitable to capture preferences from a small group of decision-makers and stakeholders instead of capturing individual preferences across the whole population [95]. Another problem regards the quantification of criteria since it forces participants to be explicit about how they evaluate the criteria, such as evaluating the effectiveness of a drug and its associated harms and the degree of uncertainty regarding benefits and risks [96]. This type of assessment may widely vary since there is a personal evaluation of the importance and severity of health states associated with the disease or the treatment plan, leading the modelling approaches to make mistakes in determining patients' preferences.

Finally, although this type of model is now widely used in case studies and shows a reliable application for conflict situations such as found in multimorbidity clinical cases, it has not yet applied in the clinical facilities. In more extensive and complex situations such as those found in the real clinical environment where multiple alternative recommendations are conflicting, it is not easy to evaluate these methods' application. However, until now, the feedback to the state of art case studies has been quite positive.

Chapter 3

Publications Composing the Doctoral Thesis

This chapter provides the publications selected to describe the work developed. By presenting these articles, we are aimed at addressing the different areas of this doctoral work and the objectives outlined in section 1.6. At the beginning of each section, a table summarises the content of each article.

3.1 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 (2016)	0.538, Health Informatics (Q2), Health Information Management (Q2), Information Systems (Q2)

Contribution of the doctoral candidate
The doctoral candidate, António José Linhares da Silva, declares to be one of the main authors and a significant 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

Decision Support Provided by a Temporally Oriented Health Care Assistant

An Implementation of Computer-Interpretable Guidelines

Tiago Oliveira · António Silva · José Neves · Paulo Novais

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.

Keywords Clinical Decision Support · Computer-Interpretable Guidelines · Ontologies · · Temporal Representation · Reminder Systems

1 Introduction

Clinical Decision Support Systems (CDSSs) may be classified in different categories according to the roles they perform. In [1], a broad classification is proposed featuring the following categories: *tools for information management*, *tools for focusing attention*, and *tools for patient-specific recommendations*. The first includes information retrieval systems that provide the data and knowledge needed by physicians, whereas the second concerns systems that alert or notify health care professionals of situations that need their attention. The third, and last, category consists of systems providing custom-tailored assessments of patients based on patient data. The ultimate goal of all these

T. Oliveira · A. Silva · J. Neves · P. Novais
University of Minho, Braga, Portugal
E-mail: {toliveira,jneves,pjon}@di.uminho.pt, pg25307@alunos.uminho.pt

types of systems is to guarantee that the clinical process is following a correct path and there is the best possible outcome in terms of patient state, through the provision of decision support. Although exhaustive proof of the advantages of the widespread use of CDSSs is lacking, there is isolated evidence of outcome improvements brought on by these systems in specific settings [2; 3; 4]. Current challenges in CDSS development are mainly concerned with making these systems user-centric and making them easily accessible over the internet by prioritizing and filtering the recommendations that are presented to users at a given time and place [5].

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. The present work proposes a solution in the form of a CDSS that gathers such functionalities, supported by machine-readable versions of Clinical Practice Guidelines (CPGs). The use of Computer-Interpretable Guidelines (CIGs) [6; 7] endows the system 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 underlying model for CIGs is formalized in Ontology Web Language (OWL) [8], 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 five sections. Section 2 contains a description of the main existing models and tools for the temporal representation and execution of CIGs, their strengths, and their limitations. Section 3 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 4 presents conclusions about the work developed so far and future work considerations.

2 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 [9; 10; 11; 12; 13]. . 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.

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 [13], which provides representation primitives only for one clinical recommendation, all the other models allow the definition of networks of clinical tasks. It is possible to observe that the duration and waiting time patterns are present in most models, the exception being the Guideline Interchange Format (GLIF3) [11], which does not provide a constructor to express delays to the starting times of tasks. When it comes to repetition conditions, only PROforma [10] and the GuideLine Acquisition, Representation and Execution (GLARE) [12] offer sufficiently expressive constructors for this temporal pattern. In fact, the GLARE [12] model is also specialized in the representation of periodic events and enables the definition of nested periodicities. In terms of temporal expressiveness, it is the most complete. Periodicities are the most important pattern as they are used to define the frequency of treatments, exams, and general check-ups. Another model that offers a comprehensive representation of this pattern is Asbru [9]. Table 1 shows that each model has at least one limitation in one type of temporal constraint. A common drawback of current CIG models is that they do not provide adequate representation primitives for temporal constraints regarding conditions about the state of a patient.

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) [14], SAGEDesktop [15], or the execution engine of GLARE [16]. However, these tools are limited and focus manly 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 [6].

Table 1 The symbol ✓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 [13]	✓	✗	✗	✓	✗	✗
GLIF3 [11]	✓	✗	✗	✗	✗	✓
Asbru [9]	✓	✓	✓	✓	✗	✗
PROforma [10]	✓	✓	✗	✓	✓	✗
GLARE [12]	✓	✓	✓	✓	✓	✗

3 Development of the Health Care Assistant

The solution proposed for the challenges mentioned in Section 1 is supported by an architecture such as the one of the CompGuide system, represented in Fig. 1. It 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. The sections below provide descriptions of its architecture, underlying CIG representation, temporal model, and web-based health care assistant tool. The highlight will be placed on the temporal aspects of CIGs, whose interpretation enables the creation of a calendar for guideline execution.

3.1 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 interpretation.

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 *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 [17], aims to facilitate the integration of CIGs into any type of application and make them widely accessible, thus enabling differently oriented implementations. The *Health Care Assistant* (HCA) is one such implementation.

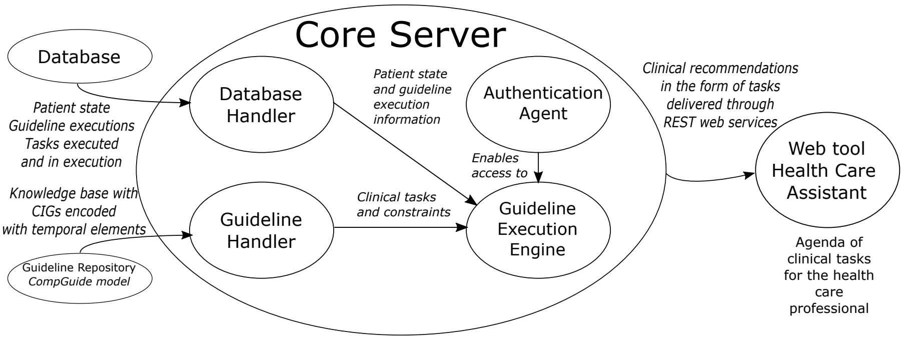

Core Server
Database
Patient state
Guideline executions
Tasks executed
and in execution
Database Handler
Patient state and guideline execution information
Authentication Agent
Enables access to
Knowledge base with CIGs encoded with temporal elements
Guideline Handler
Clinical tasks and constraints
Guideline Execution Engine
Guideline Repository
CompGuide model
Clinical recommendations in the form of tasks delivered through REST web services
Web tool Health Care Assistant
Agenda of clinical tasks for the health care professional

Fig. 1 Architecture of the CompGuide system with the main elements and functions of its *Core Server*.

3.2 Representation of Computer-Interpretable Guidelines

Before the definition of a temporal model, it is necessary to have an underlying structure for the representation of essential CIG elements. The CompGuide model [18] offers constructors in OWL for the definition of different types of clinical tasks, the relative order of tasks, and clinical constraints placed on tasks. This ontology language, and more specifically its description logics version, was chosen due to the completeness of its description of different entities and relationships between them. The main types of clinical tasks defined in CompGuide are represented with the following classes:

- *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 formalization in CompGuide resembles a linked list of recommendations, which, when interpreted, becomes a clinical careflow. 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.

3.3 Temporal Elements of Guidelines

Fig. 2 shows diagram of the classes representing temporal elements in the ontology. The classes that enable the representation of temporal restrictions are all subclasses *TemporalElement*. 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.

3.3.1 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*

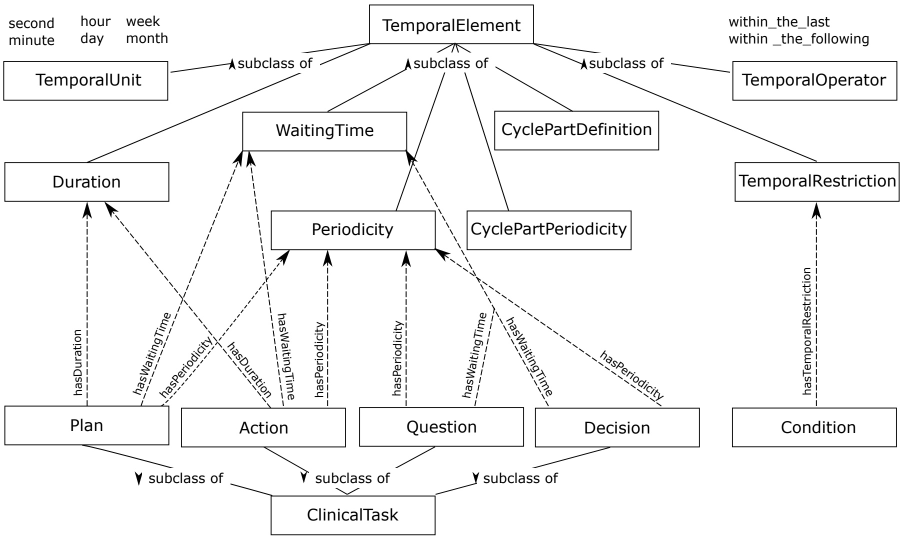


Fig. 2 Classes of the temporal model used in CompGuide.

object property, which connects them with 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).

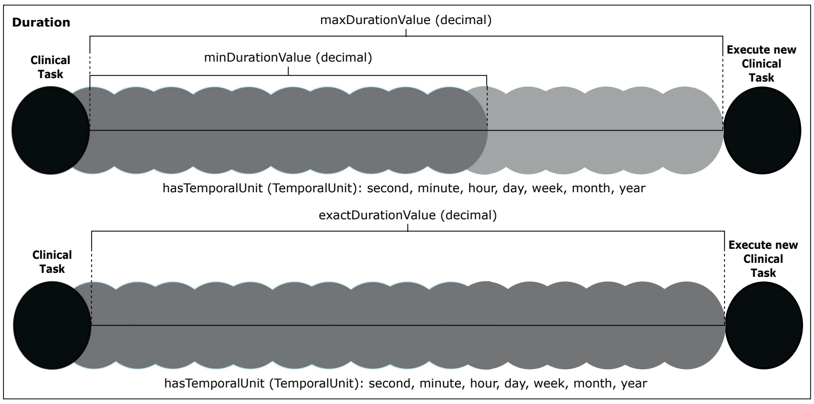

Duration
maxDurationValue (decimal)
minDurationValue (decimal)
Clinical Task
Execute new Clinical Task
hasTemporalUnit (TemporalUnit): second, minute, hour, day, week, month, year
exactDurationValue (decimal)
Clinical Task
Execute new Clinical Task
hasTemporalUnit (TemporalUnit): second, minute, hour,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

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

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.

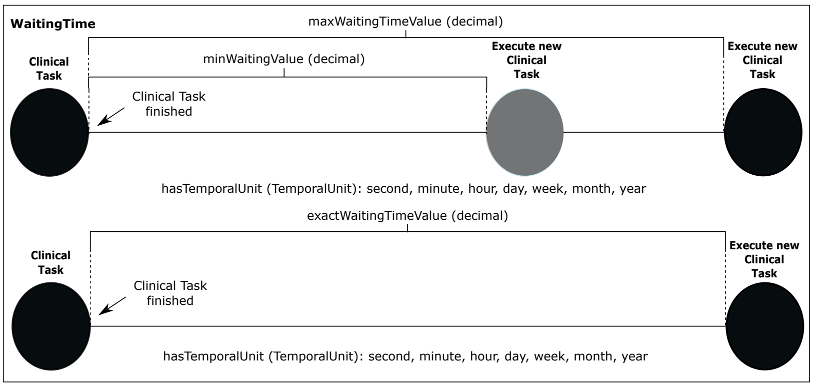

WaitingTime
maxWaitingTimeValue (decimal)
Clinical Task
minWaitingValue (decimal)
Execute new Clinical Task
Execute new Clinical Task
Clinical Task finished
hasTemporalUnit (TemporalUnit): second, minute, hour, day, week, month, year
exactWaitingTimeValue (decimal)
Clinical Task
Execute new Clinical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

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

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. , 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 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 with a *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.

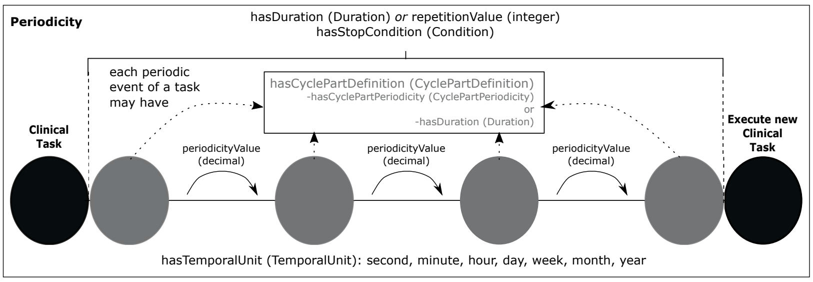

Periodicity
hasDuration (Duration) or repetitionValue (integer)
hasStopCondition (Condition)
each periodic
event of a task
may have
hasCyclePartDefinition (CyclePartDefinition)
-hasCyclePartPeriodicity (CyclePartPeriodicity)
or
-hasDuration (Duration)
Clinical
Task
periodicityValue
(decimal)
periodicityValue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

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

3.3.2 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, pre-conditions and conditions of rules in *Decision* instances. 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.

3.4 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) [19] 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. Figs. 6 and 7 show the instantiation of case examples for each temporal pattern. As can be seen, 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 same is to say 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 and 4 from Fig. 6 and Case 5 from Fig. 7 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 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.

When comparing with the approaches mentioned in Table 1, the examples having periodicities would not have been represented in at least three of the models. Additionally, the CompGuide model provides a set of representation primitives for the representation of temporal constraints on conditions about the state of the patient, which is something that the most comprehensive temporal model in the literature, GLARE [12] does not contemplate.

3.5 Web-based Tool for the Visualization and Execution of Guidelines

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 tasks, 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.

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 3.4 can be seen as side messages, as shown in 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.

4 Conclusions and Future Work

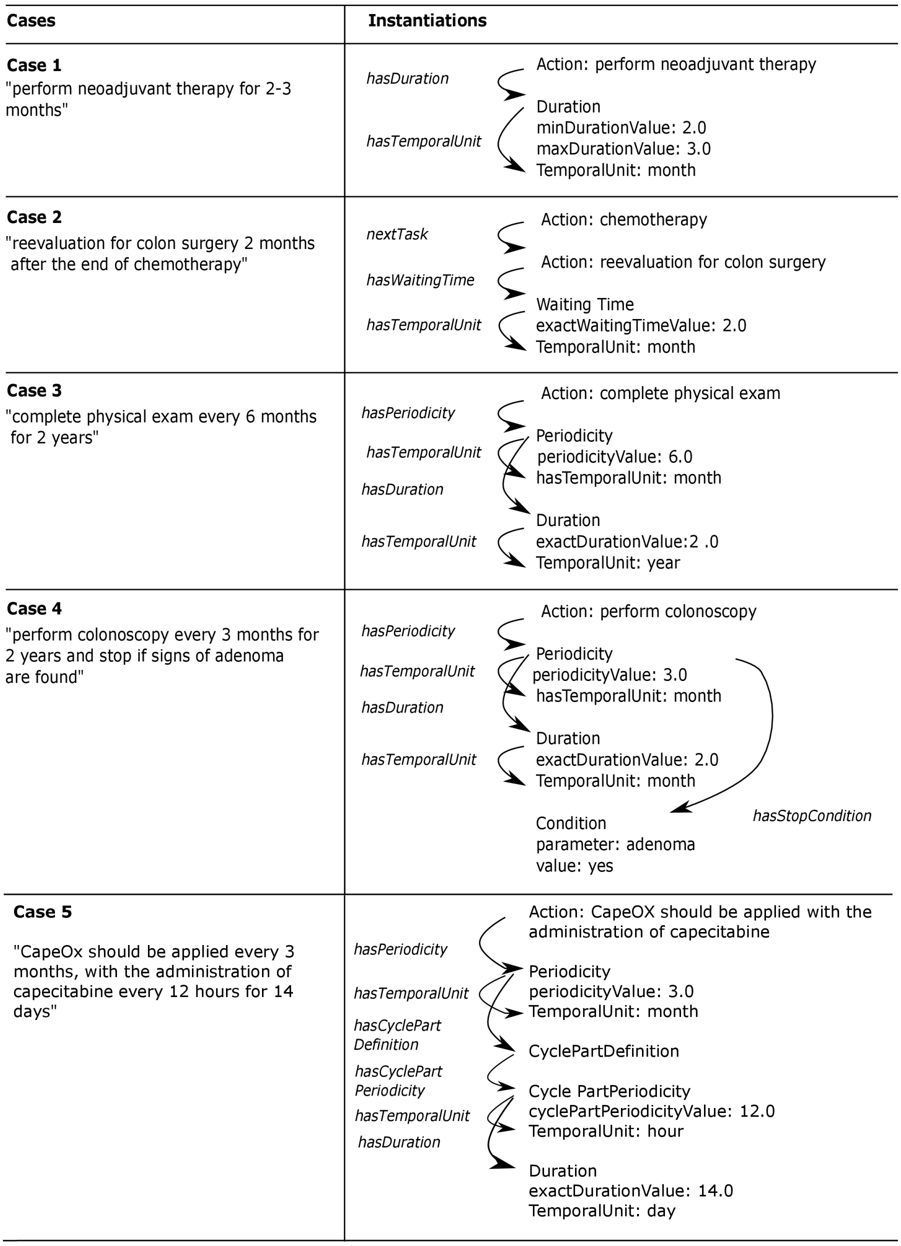
The work presented herein is an example of an implementation of the notion of guidelines as services, presented in [20], which takes advantage of the flexibility of the CompGuide system. The main contributions are a comprehensive temporal representation model and a web-based tool for the execution of 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 health care professionals and help them 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.

As future work, it is necessary to evaluate the HAC tool by performing usability tests with health care professionals. 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 within the HAC, but also on their own professional calendars, with the other events of their daily practice, outside the scope of guideline execution.

References

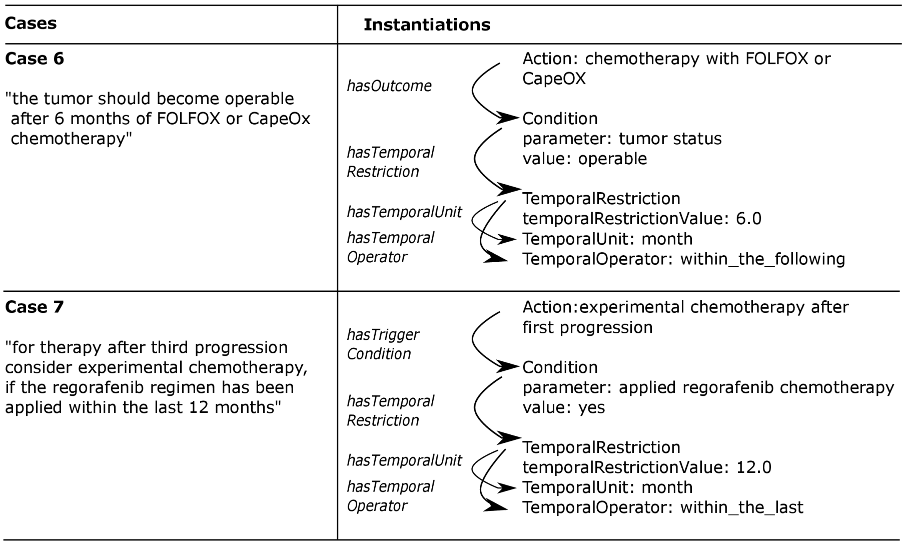
1. M.A. Musen, Y. Shahar, E.H. Shortliffe, in *Biomedical Informatics*, ed. by E. Shortliffe, J. Cimino, Health Informatics (Springer New York, 2006), pp. 698–736. DOI 10.1007/0-387-36278-920
2. R. Kaushal, K.G. Shojania, D.W. Bates, Archives of internal medicine **163**(12), 1409 (2003). DOI 10.1001/archinte.163.12.1409
3. N.K.J. Adhikari, J. Beyene, J. Sam, R.B. Haynes, Journal of the American Medical Association **293**(10), 1223 (2005)
4. T. Bright, A. Wong, ..., Annals of Internal Medicine **157**(1), 29 (2012). DOI 10.7326/0003-4819-157-1-201207030-00450
5. D.F. Sittig, A. Wright, J.a. Osheroff, B. Middleton, J.M. Teich, J.S. Ash, E. Campbell, D.W. Bates, Journal of biomedical informatics **41**(2), 387 (2008). DOI 10.1016/j.jbi.2007.09.003
6. D. Isern, A. Moreno, International Journal of Medical Informatics **77**(12), 787 (2008). DOI 10.1016/j.ijmedinf.2008.05.010
7. M. Peleg, Journal of Biomedical Informatics **46**(4), 744 (2013). DOI 10.1016/j.jbi.2013.06.009
8. D.L. McGuinness, F. Van Harmelen. OWL Web Ontology Language Overview (2004). URL `https://www.w3.org/TR/owl-features/`
9. Y. Shahar, S. Miksch, P. Johnson, Artificial intelligence in Medicine **14**(1-2), 29 (1998). DOI 10.1016/S0933-3657(98)00015-3
10. A. Vollebregt, A. ten Teije, F. van Harmelen, J. van der Lei, M. Mosseveld, Artificial Intelligence in Medicine **17**(2), 195 (1999). DOI 10.1016/S0933-3657(99)00016-0
11. A.a. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q.T. Zeng, D. Wang, V.L. Patel, R.a. Greenes, E.H. Shortliffe, Journal of Biomedical Informatics **37**(3), 147 (2004). DOI 10.1016/j.jbi.2004.04.002
12. L. Anselma, P. Terenziani, S. Montani, A. Bottrighi, Artificial Intelligence in Medicine **38**(2), 171 (2006). DOI 10.1016/j.artmed.2006.03.007

13. M. Samwald, K. Fehre, J. de Bruin, K.P. Adlassnig, Journal of biomedical informatics (2012). DOI 10.1016/j.jbi.2012.02.001
14. D. Wang, M. Peleg, S.W. Tu, A.A. Boxwala, O. Ogunyemi, Q. Zeng, R.A. Greenes, V.L. Patel, E.H. Shortliffe, Journal of Biomedical Informatics **37**(5), 305 (2004). DOI 10.1016/j.jbi.2004.06.002
15. D. Berg, P. Ram, J. Glasgow, J. Castro, The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society **4**, 3217 (2004). DOI 10.1109/IEMBS.2004.1403906
16. P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G. Molino, G. Correndo, Studies in Health Technology and Informatics **101**(3), 162 (2004). DOI 10.3233/978-1-60750-944-8-162
17. T. Oliveira, P. Leão, P. Novais, J. Neves, in *Trends in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection SE - 18*, *Advances in Intelligent Systems and Computing*, vol. 293, ed. by J. Bajo Perez, J.M. Corchado Rodríguez, P. Mathieu, A. Campbell, A. Ortega, E. Adam, E.M. Navarro, S. Ahrndt, M.N. Moreno, V. Julián (Springer International Publishing, 2014), pp. 149–156. DOI 10.1007/978-3-319-07476-4{_}18
18. T. Oliveira, P. Novais, J. Neves, in *Trends in Practical Applications of Agents and Multiagent Systems SE - 10*, *Advances in Intelligent Systems and Computing*, vol. 221 (Springer International Publishing, 2013), pp. 77–85
19. A. Benson, T. Bekaii-Saab, E. Chan, Y.J. Chen, M. Choti, H. Cooper, P. Engstrom, NCCN Clinical Practice Guideline in Oncology Colon Cancer. Tech. rep., National Comprehensive Cancer Network (2013). URL `http://www.nccn.org/professionals/physician_gls/f_guidelines.asp`
20. P. Novais, T. Oliveira, J. Neves, Progress in Artificial Intelligence pp. 1–7 (2016). DOI 10.1007/s13748-016-0084-2. URL `http://dx.doi.org/10.1007/s13748-016-0084-2`



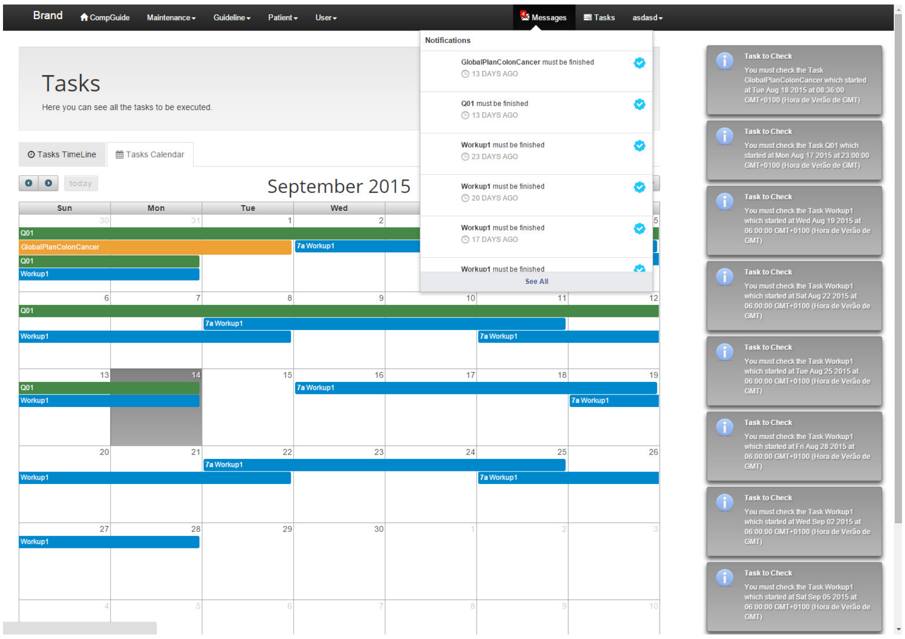
Cases	Instantiations
Case 1 "perform neoadjuvant therapy for 2-3 months"	Action: perform neoadjuvant therapy *hasDuration* Duration minDurationValue: 2.0 maxDurationValue: 3.0 *hasTemporalUnit* TemporalUnit: month
Case 2 "reevaluation for colon surgery 2 months after the end of chemotherapy"	Action: 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*.



Cases	**Instantiations**	
Case 6 "the tumor should become operable after 6 months of FOLFOX or CapeOx chemotherapy"	*hasOutcome* *hasTemporal Restriction* *hasTemporalUnit* *hasTemporal Operator*	Action: chemotherapy with FOLFOX or CapeOX Condition parameter: tumor status value: operable TemporalRestriction temporalRestrictionValue: 6.0 TemporalUnit: month 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"	*hasTrigger Condition* *hasTemporal Restriction* *hasTemporalUnit* *hasTemporal Operator*	Action:experimental chemotherapy after first progression Condition parameter: applied regorafenib chemotherapy value: yes TemporalRestriction temporalRestrictionValue: 12.0 TemporalUnit: month TemporalOperator: within_the_last

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



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

See All

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)

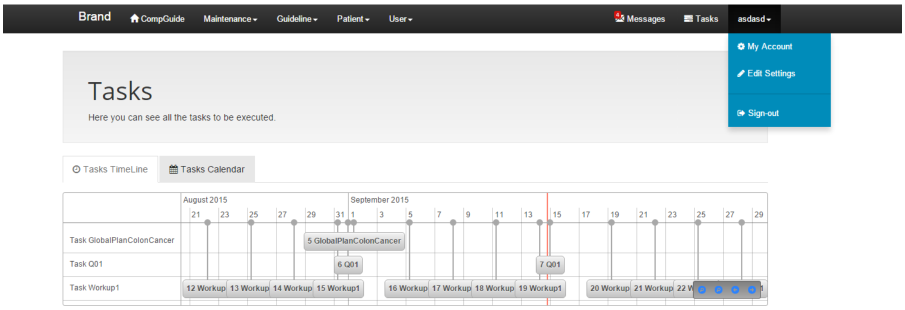
Task to Check
You must check the Task Workup1 which started at Tue Aug 25 2015 at 06:00:00 GMT+0100 (Hora de Verão de GMT)

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)

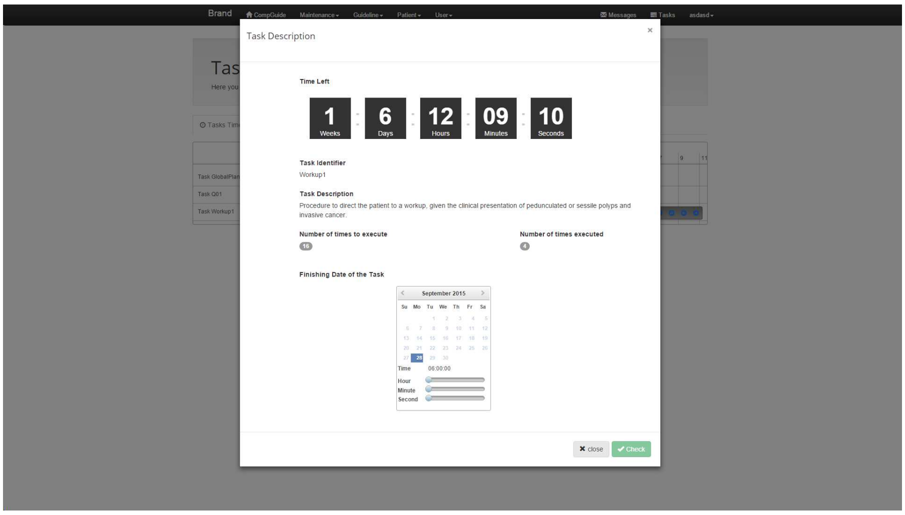
Fig. 8 Calendar task view and notifications of the Health Care Assistant.



Tasks

Here you can see all the tasks to be executed.

Fig. 9 Timeline task view of the Health Care Assistant.



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

Time 06:00:00

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

3.2 A System for the Management of Clinical Tasks Throughout the Clinical Process with Notification Features

Title	A system for the management of clinical tasks throughout the clinical process with notification features
Authors	António Silva, Tiago Oliveira, José Neves, Ken Satoh, and Paulo Novais
Publication Type	Conference paper
Chapter	Agents and Multi-Agent Systems for Health Care
Book series	Lecture Notes in Computer Science
Publisher	Springer, Cham
Volume	10685
Pages	76-93
Year	2017
Online ISBN	978-3-319-70887-4
DOI	https://doi.org/10.1007/978-3-319-70887-4_5
URL	http://link.springer.com/article/10.1007/s10916-016-0655-6
State	Published online
Scimago journal rank (2017)	0.295, Computer Science Miscellaneous (Q2)

Contribution of the doctoral candidate
The doctoral candidate, António José Linhares da Silva, declares to be the main author and the major contributor of the paper *A system for the management of clinical tasks throughout the clinical process with notification features*.

Lecture Notes in Computer Science
DOI 10.1007/978-3-319-70887-4_5

A System for the Management of Clinical Tasks Throughout the Clinical Process with Notification Features

António Silva, Tiago Oliveira, José Neves,
Ken Satoh, Paulo Novais

Accepted: 25 November 2017

Abstract Computer-Interpretable Guidelines have been associated with a higher integration of standard practices in the daily context of health care institutions. The Clinical Decision Support Systems that deliver these machine-interpretable recommendations usually follow a Q&A style of communication, retrieving information from the user or a clinical repository and performing reasoning upon it, based on the rules from Clinical Practice Guidelines. However, these systems are limited in the reach they are capable of achieving as they were initially conceived for use in very specific moments of the clinical process, namely in physician appointments. The purpose of this work is thus to present a system that, in addition to Q&A reasoning, is equipped with other functionalities such as the scheduling and temporal management of clinical tasks, the mapping of these tasks onto an agenda of activities to allow an easy consultation by health care professionals, and notifications that let health care professionals know of task enactment times and information collection times. In this way, the system ensures the delivery of procedures. The main components of the system, which reflect a different perspective on the delivery of CIG advice that we call guideline as a service, are disclosed, and they include a health care Personal Assistant Web Application, a health care assistant mobile application, and the integration with the private calendar services of the user.

A. Silva
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: asilva@algoritmi.uminho.pt

T. Oliveira · K. Satoh
National Institute of Informatics, Tokyo, Japan
E-mail: {toliveira, ksatoh}@nii.ac.jp

J. Neves · P. Novais
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: {jneves,pjon}@di.uminho.pt

1 Introduction

Computer-Interpretable Guidelines (CIGs) are machine-interpretable versions of Clinical Practice Guidelines (CPGs). The latter are systematically developed statements associated with the promotion of best medical practices and reduction of medical error [1]. The aim of these documents is to provide clinical advice for specific circumstances and to support health care professionals in their decisions [2]. Their formalisation as CIGs in Clinical Decision Support Systems (CDSSs) brings forth the development of a new range of operations that can be performed with the knowledge they enclose. Such include automated reasoning for the generation of recommendations, consistency checking within the same CIG and across different CIGs, and merging CIG knowledge with contextual information such as patient and physician preferences or available health care resources, to name a few [3]. The point of these operations is to tailor care in order to generate better outcomes and avoid adverse events. Nonetheless, managing patients is a challenging endeavour for health care professionals given that they are typically responsible for numerous cases at the same time and each case involves the enactment of several and complex procedures. Managing this complexity is something that the current applications for CIG execution do not contemplate in the functionalities they offer [4–7]. Current CIG-based systems do not provide mechanisms for integrating CIG recommendations in the daily routine of health care institutions, which calls forth the need for such systems to assume a new style of communication that can further promote a positive impact on the outcomes of care [8].

CIGs are considered to be the best approach to the concept of *living guidelines*, which captures statements for clinical decision support that are dynamic - in the sense that they are capable of evolving and providing advice based on the latest evidence - and interactive [9]. This interactive component is related to the ability to cover tasks such as patient tracking, patient follow-up, scheduling of procedures, and the monitoring of procedure constraints, and, at the same time, autonomously inform health care professionals about important aspects of these procedures in the most diverse situations.

Following the identified need for different ways in which to deliver CIG-based advice, the work herein proposes a different perspective regarding this matter. Its main contribution is a system that allows different implementations of CIGs. We show how these implementations can be differently oriented through a Personal Assistant Web Application and a health care assistant mobile app. The principle behind the system and the presented implementations is that the constraints supported my a CIG model and placed on clinical tasks can be used to enhance CIG-based CDSSss. The system is based on the CompGuide ontology for CIGs [10], which treats CPGs as sets of various clinical tasks organised in a work flow. The present paper represents an extension of the work in [11].

Section 2 describes related work regarding systems for CIG execution, featuring a description of their means of operation. In Section 3, we present the CompGuide ontology and respective main features that led to the implementations described in the following section. Section 4 provides details about the CompGuide architecture for the deployment of CIGs and how its services are used as a basis for the Personal Assistant Web Application and the health care assistant mobile application developed to accompany health care professionals. Section 5 describes the functionalities supporting care with examples of CIG execution. Finally, Section 6 presents the conclusions drawn so far with the development of the health care assistant and future directions for the work.

2 Existing Systems for CIG Execution

Based on the classification of CDSSs presented in [8] - tools for information management, tools for patient-specific advice, and tools for focusing attention - and the analysis of current CIG execution approaches [12], it is possible to observe that the most significant examples fall under the category of tools for patient-specific advice. This is the case of CIG execution engines such as the GLIF3, Guideline Execution Engine (GLEE) [4], the Spock Engine [5], and the GLARE Execution Engine [6], which were specifically developed for the application of guidelines to patients in health care settings.

GLEE [4] is based on the third version of the Guideline Interchange Format (GLIF3) [13], which, in turn, was designed to support guideline modelling as a flowchart of structured steps that represented clinical actions and decisions. The architecture of GLEE provides three layers of *abstraction*, namely *data*, *business logic* and *user interface*. The *data* layer contains an electronic medical record with patient data, a guideline repository, and a clinical event monitor that allows the execution of CIGs driven by clinical events. The *business logic* layer contains an execution engine consisting of a server and many clients that interact with users. The bottom interface layer contains the applications responsible for exchanging information with the upper layers. The execution engine records every clinical parameter from a patient during the execution of a CIG, suggesting actions to be performed. In addition, the user can control the process by confirming or deciding different transitions between actions.

The Spock Engine [5] was developed to enable the execution of CIGs in the Asbru model [14]. It incorporates an inference engine that can retrieve data from the patient's electronic medical record. 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. This execution engine stores different data structures such as state transitions, a queue of scheduled awaiting tasks, and the list of recommendations already issued. This strict control of tasks stems from the expressiveness of the Asbru temporal model, which provides various temporal patterns for the control of recommendation steps.

The GLARE Execution Engine was also developed based on a CIG model focused on temporal constraints, the Guideline Acquisition, Representation and Execution (GLARE) [6]. CIGs in GLARE follow 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. Similarly to the other systems, GLARE also defines three abstraction layers. In this case they are called *system*, *xml*, and *dbms*. The system layer contains an execution interface tied to an execution engine that interprets and updates XML files representing instances of patients and guideline executions in the *xml* layer. These are intermediate structures used to exchange data with the *dbms* layer and the *system* layer in a structured way.

All these systems use structures and well-defined languages that can be read and analysed by a program. Furthermore, they also feature a guideline repository, a run-time engine for the CIG knowledge, and an electronic medical record. Furthermore, they may, as in the case of the GLARE Execution Engine, support modules that describe the context, mainly in terms of available resources, of the health care institution where CIG deployment is taking place. Their objective is to run CIG instructions against data from patients and produce tailored recommendations, according to the observed state. In these systems, the role of the execution engine is straightforward, in the sense that it is merely concerned with following the constraints of the clinical work flow, comparing items of the patient state with conditions stated in rules dictating whether a recommendation should be provided or not. Most applications for CIG execution, including the above-given examples, exist in the form of client-server applications, with the intelligence engine placed on the client side. Furthermore, these applications are mostly available as desktop applications, which is an obstacle to their potential for reaching health care professionals and their ease of deployment.

The idea of enhancing CDSSs with additional features that allow them to achieve a higher level of integration of clinical recommendations in clinical practice comes from the ever-increasing role of Ambient Assisted Living (AAL) in enabling new information and communication services which transparently support people in their everyday lives [15,16]. In fact, a similar idea has been explored in [17], where a personal memory assistant, capable of intelligent scheduling and deployed over a platform, called iGenda. The assistant acts as the support for a centralised manager system that can manage several services and is responsible for the scheduling of multiple agendas, taking into account the availability of resources or the health conditions of the users. Although different, the work proposed herein can be related to this project and others such as the Collaborative Memory Aids [18] and Hermes [19], but with the focus placed on the health care professional.

3 CompGuide Ontology for Clinical Practice Guidelines

The CIG model used in this work is the CompGuide ontology [10]. It provides representation primitives for clinical recommendations based on Web Ontology Language (OWL) by following a task network model in which each recommendation assumes the form of a task. In order to reflect this, a set of key OWL classes were defined as subclasses of *ClinicalTask*. They include the folllowing:

- *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, featuring various options and respective conditions;
- *Plan*: a composed task containing instances of the other tasks defined to achieve a specific goal.

In CompGuide there are object properties that connect instances of the classes as mentioned above in order to define the relative order between tasks. 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, conditions for options in *Decision* tasks, and expected outcomes for clinical tasks. The *Condition* class allows the representation of these conditions with specific properties for clinical parameters and their values.

The classes that enable the representation of temporal restrictions are all subclasses of *TemporalElement* [20]. The relationship between these temporal classes and the classes in ClinicalTask are shown in Figure 1, along with the properties used to connect them. One of the subclasses of *TemporalElement* 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 main classes that enable the definition of temporal restrictions about the execution of tasks are:

- *Duration*: definition of how long *Actions* and *Plans* should last.
- *WaitingTime*: definition of a delay in the start of a clinical task.
- *Periodicity*: definition of the frequency of a clinical task.
- *CyclePartPeriodicity*: a nested temporal pattern for the definition of a periodicity within a periodicity.

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. *TemporalOperator* consists of two instances, *within_the_last* and *within_the_following*. The operator *within_the_last* is used when one aims 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, pre-conditions

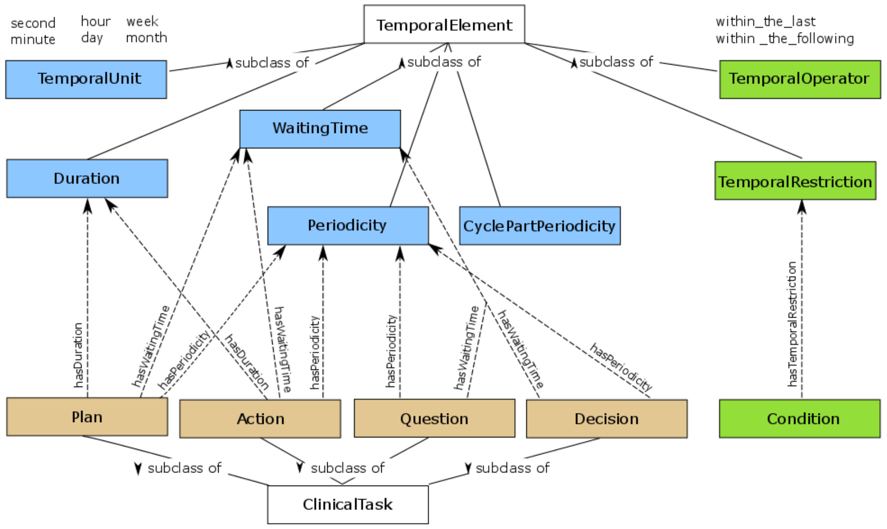


Fig. 1: Representation of the CompGuide ontology with clinical tasks and respective temporal elements.

and conditions of rules in *Decision* instances. This operator is interpreted 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. Within the context of a CPG for the diagnosis and treatment of colon cancer, an example of a temporal restriction would be an *Action* that advised chemotherapy with an outcome stating that the tumour should become operable within six months. In this case, there is a condition with a temporal restriction featuring a *within_the_following* operator.

The details of the CompGuide model are further provided in [20], along with an assessment of the expressiveness of the model compared to other approaches that revealed that it enables the representation of more temporal patterns. The interpretation of the work flow of tasks, their clinical constraints, and their temporal constraints demands an execution engine capable of analysing these three aspects and crossing them with patient information. However, these instructions may become too intricate for a clear understanding, which demands ways of delivering CIGs that also help to manage the complexity of these recommendations during their enactment.

4 CompGuide Architecture for CIG Execution

The CompGuide system follows a service-oriented architecture that aims to provide recommendations to support medical decision-making. As shown in Figure 2, it consists of a *Core Server* that is the central component of the architecture and was developed as a RESTful web service application. The usage of web services as the means to access the *Core Server* offers consistent performance to access the web resources, better scalability and modifiability, providing the possibility of improving selected services without compromising others. This architectural style grants greater flexibility when integrating CIG execution functionalities in third party applications [21]. Given the architecture style used for the system and the concept of a centralised CIG management system that allows different implementations, the distribution of CompGuide follows a *software* as a service (Saas) model.

The *Core Server* has four modules: the *Authentication Agent*, the *Guideline Handler*, the *Database Handler* and the *Guideline Execution Engine*. The *Authentication Agent* is the component responsible for the authentication and authorization of the different types of users of the system, namely administrators and health care professional, such as physicians or nurses. The *Guideline Handler* is responsible for managing the access to recommendations of CIGs in the *Guideline Repository*, keeping different CIGs represented according to the CompGuide ontology, organised by authorship and by date. This component consists of a collection of OWL files. In order to use a CPG for execution, the *Guideline Handler* accesses the selected CIG in the *Guideline Repository* and pulls the corresponding care flow, delivering it to the *Guideline Execution Engine*. This module uses information about the patient state provided by the *Database Handler* as well as temporal constraints on the execution of the clinical tasks and temporal constraints on the state of a patient given by the *Guideline Handler* to fill in the data entry points of the care flow and produce recommendations. Thus, the *Guideline Execution Engine* interprets all the scheduling constraints on the tasks and produces enactment times. The applications implemented to interact with the health care professionals are then responsible for verifying starting and ending. These mechanisms to follow the execution of procedures over time and to check the execution of tasks are absent from most CIG frameworks [22], but they are essential to have a decision support that is truly capable of following up on guideline deployment.

The *Core Server*, as mentioned before, provides these features as RESTful web services implemented in Java, using the RESTEasy API over a WildFly Application Server. The Personal Assistant Web Application, which uses the web services available in the *Core Server*, was developed as a web application following the Model-View-Control(MVC) paradigm using Java Server Faces (JSF). The *Health Care Assistant Mobile Application* is an android application developed in Java, which also uses the same web services. The purpose of the *Core Server* is to make available CIG services that anyone can integrate into their own applications, with a special focus on AAL applications. Following

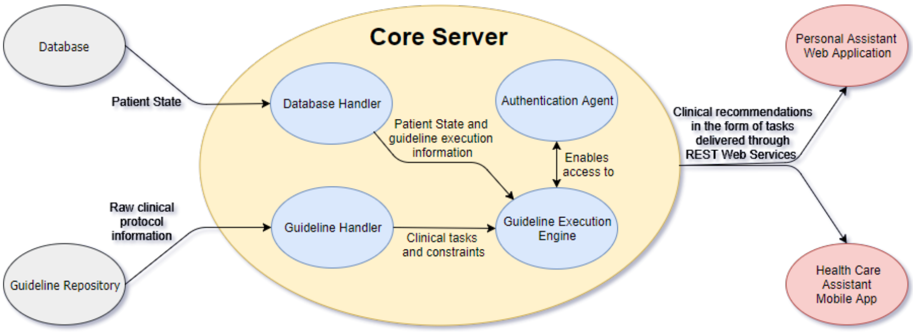

Database
Patient State
Core Server
Database Handler
Authentication Agent
Patient State and guideline execution information
Enables access to
Raw clinical protocol information
Guideline Handler
Clinical tasks and constraints
Guideline Execution Engine
Guideline Repository
Clinical recommendations in the form of tasks delivered through REST Web Services
Personal Assistant Web Application
Health Care Assistant Mobile App

Fig. 2: Architecture of CompGuide system

the parallel with Saas, this form of delivering CIGs can be considered to be *guideline as a service*.

4.1 CompGuide RESTful Web Services

The CompGuide web services provide a set of features that allows accessing the *Guideline Repository* as well as saving, removing and updating information in the *Database*. Their description is as follows.

The *Guideline Service* handles the logic of the execution of a guideline, task to task, obtaining codified tasks in the ontology, providing them as recommendations. The *Get Tasks Service* provides a list of tasks that must be executed at a given moment. In order to get the next task to be executed, the user must perform a request to the *Next Task Service*.

The *Guidelines Service* has only one web service that provides the list of existing guidelines in the data base. Additionally, the *Guideline Execution Service* represents the execution of a guideline initiated by a physician and associated with a patient, so this web service provides information about the execution of a guideline. To add a new execution, the user must perform a request to the *Add Guideline Execution Service*. Regarding the *Guideline Execution Active Service*, this web service provides a list of the active executions of guidelines for a specific user.

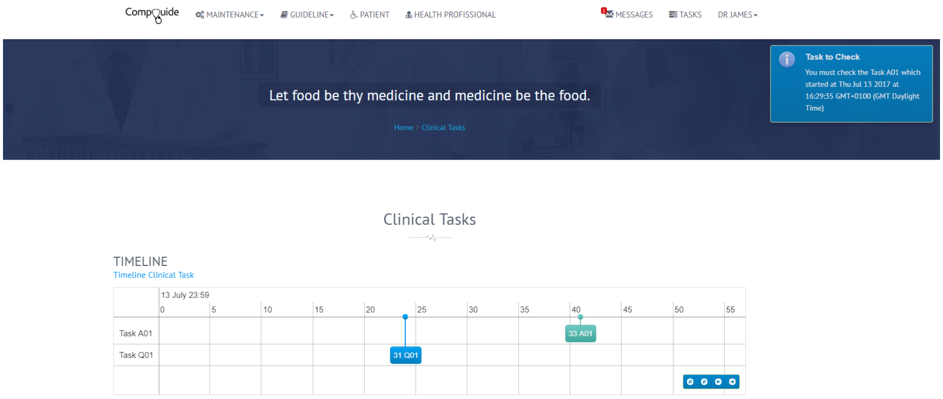
It is also possible to retrieve and alter patient information through the *Patient Service*, which allows to add, remove, update and retrieve patient information.

Finally, the *Task Service* and *User Service* follow the same structure of the previous services, allowing the access and manipulation of information about these respective entities in the *Database*.

4.2 Personal Assistant Web Application

The *Personal Assistant Web Application* is an application that highlights the role of CPGs as patient management and following tools. Based on the information provided by the *Execution Engine*, it can keep track of clinical tasks that should be carried out by the health care professional. By using information and communication systems, it is possible to provide CIGs with dynamism, presence, and interactivity that may bring them closer to the concept of living guidelines. It enables the management of information about CPGs, health care professionals that are users in the system, and patients to which CPGs are applied. As such, one can create, edit and delete all this information, according to the type of authorization in the system.

In order to facilitate the visualisation of the clinical tasks, for the health care professionals, the application provides two forms of displaying these recommendations. The first is a timeline in which all the clinical tasks are shown over a chronogram. A timeline of activities has the ability to compress multiple tasks into a single continuity without compromising the succession, and the easy understanding of clinical procedures. The benefits from such a representation include the capacity to sequence events and reduce the potential for overburdening the health care professional. Additionally, by visualising all of the pieces of a guideline treatment, care providers can make more focused, effective decisions about resources and timetables. This view is shown in Figure 3. In it, it is possible to observe clinical tasks for the management of colon cancer, namely sequential workup actions to ascertain the state of the patient.



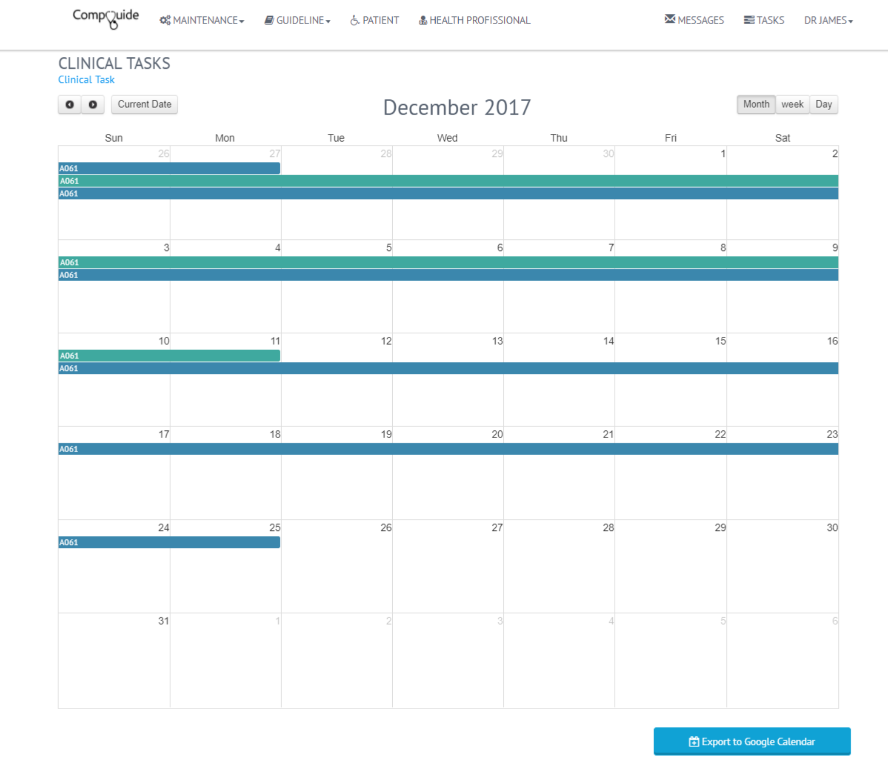
Let food be thy medicine and medicine be the food.

Task to Check

You must check the Task A01 which started at Thu Jul 13 2017 at 16:29:35 GMT+0100 (GMT Daylight Time)

Fig. 3: Timeline view of clinical procedures in the CompGuide Personal Assistant Web Application.

The other available view is a calendar in which the health care professional can visualise the tasks according to the temporal granularity he sees fit, namely week, day, and month. While with the timeline it is easier to detect the starting and ending points of tasks, with the calendar view it is easier to grasp the temporal constraints that bind clinical tasks such as durations, waiting times and periodicities. Figure 4 shows the same tasks as in the timeline, but displayed over a week, where it is possible to verify, for instance, for how long a clinical task should be applied. The purpose of the calendar view is to avoid overlooking tasks and dismissing them as that may have an adverse impact on the evolution of the patient.



CLINICAL TASKS

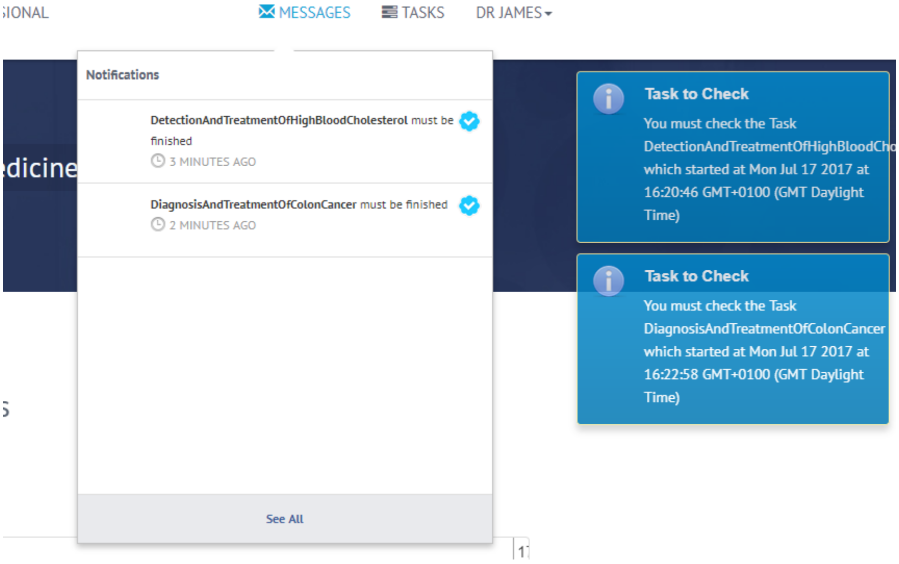
Clinical Task

December 2017

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26 A061 A061 A061	27	28	29	30	1	2
3 A061 A061	4	5	6	7	8	9
10 A061 A061	11	12	13	14	15	16
17 A061	18	19	20	21	22	23
24 A061	25	26	27	28	29	30
31	1	2	3	4	5	6

Fig. 4: Calendar view of clinical procedures in the CompGuide Personal Assistant Web Application.

In order to ensure the execution of tasks at the designated time, it was necessary to implement a notification system and a message box. These elements are both shown in Figure 5. The message box features messages such as indications about the tasks that should be performed or should have already been performed, offering the possibility to mark them as executed. As for the notification system, it is used to periodically alert the user about task enactment times and steps to collect information about the patient, such as the outcomes of clinical tasks, according to their respective temporal restrictions. The notifications are shown as a pop-up message.



Notifications

DetectionAndTreatmentOfHighBloodCholesterol must be finished
3 MINUTES AGO

DiagnosisAndTreatmentOfColonCancer must be finished
2 MINUTES AGO

Task to Check
You must check the Task DetectionAndTreatmentOfHighBloodCho which started at Mon Jul 17 2017 at 16:20:46 GMT+0100 (GMT Daylight Time)

Task to Check
You must check the Task DiagnosisAndTreatmentOfColonCancer which started at Mon Jul 17 2017 at 16:22:58 GMT+0100 (GMT Daylight Time)

Fig. 5: Message box and notification in the CompGuide Personal Assistant Web Application.

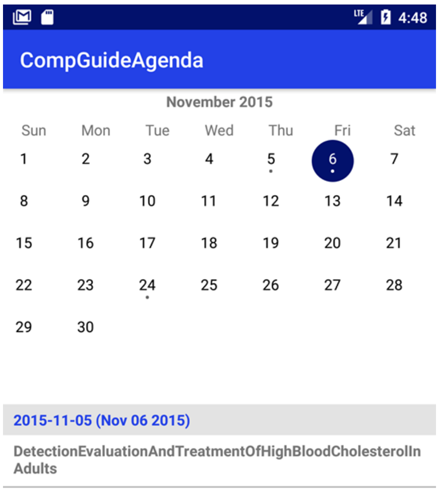
4.3 Health Care Assistant Mobile Application

In order to improve patient monitoring to increase the efficiency when treating the patients and the preparation for the appointments by the health care professionals, we developed a mobile solution. The mobile application allows the physicians or nurses to consult and monitor the progress of patients as well as the clinical recommendations wherever they need to.

The application uses the CompGuide web services to request all the patient data and clinical tasks, whereby the recommendations are displayed in a calendar of clinical procedures that was implemented using the Custom Calendar library [23]. The clinical tasks are the same that can be seen in the web application, since these two assistants, the web and mobile application, use the centralised RESTful web service developed in the *Core Server*. The fact that all the data is centralised in only one component allows a better tracking of the user actions, greater control over his decisions and get constant supply of clinical recommendations.

The calendar widget provides the view and methods necessary to display a calendar and schedule events. With this calendar, it is possible to navigate through the months and by clicking on a particular date, all the events for that day are shown below in the calendar, as depicted in Figure 6.

Its main objectives are to provide timely clinical recommendations and integrate them in the clinical practice of the health care professional. As future work, a push notification feature can be implemented in order to inform the users of when they should execute clinical tasks, when they should start them and when they should finish them.



November 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
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					

2015-11-05 (Nov 06 2015)

DetectionEvaluationAndTreatmentOfHighBloodCholesterolIn Adults

Fig. 6: Calendar view of clinical procedures in the mobile application.

4.4 Integration with Google Calendar

The Google calendar API was developed to allow the integration of applications with Google calendar and its features. The managing of events and the push notifications are the most interesting features, the user can use to monitor and supervise the clinical tasks to take control of all patient parameters and clinical process. With this API, it is possible to manage the information regarding the clinical recommendations as well as oversee and follow-up these tasks anytime and anywhere with only a mobile device. Thus, both health care assistants can sync the calendar present in CompGuide with their Google calendar account. The Google calendar provides a public RESTful API that allows the integration with a variety of devices and services on the internet. This API lets the users display, create and modify calendar events as well as work with many other calendar-related objects, such as calendars or access controls [24]. Furthermore, its Java API is native to the Android operating system, allowing a possible integration in the future with the mobile application.

Regarding the integration of the API, firstly it was necessary the registration of the application in the Google console, and then the download of Google credentials, to use in the application. After these credentials were integrated into the project, it was possible to communicate with the API. This REST API can be utilised by making explicit HTTP calls, but there are client libraries implemented in various programming languages that make the API easier to use. Thus, we used the Java client library, since the web and mobile assistants are implemented in Java.

To export the clinical tasks, presented in the calendar view of the CompGuide web assistant, it is necessary to click on the "Export to Google Calendar" button. This view is shown in Figure 7 a). After this action, the user will be redirected to the Google consent screen, asking to authorise the CompGuide application to request some user data. If the user approves, then Google gives a temporary access token that allows the application to request user data. Therefore, the CompGuide will attach the access token to the request, process all the clinical tasks and its temporal constraints, in order to create the events into the Google calendar of the user.

Through the Google calendar application, the users can see the clinical tasks and their details by clicking on the task, as shown in Figure 7 b).

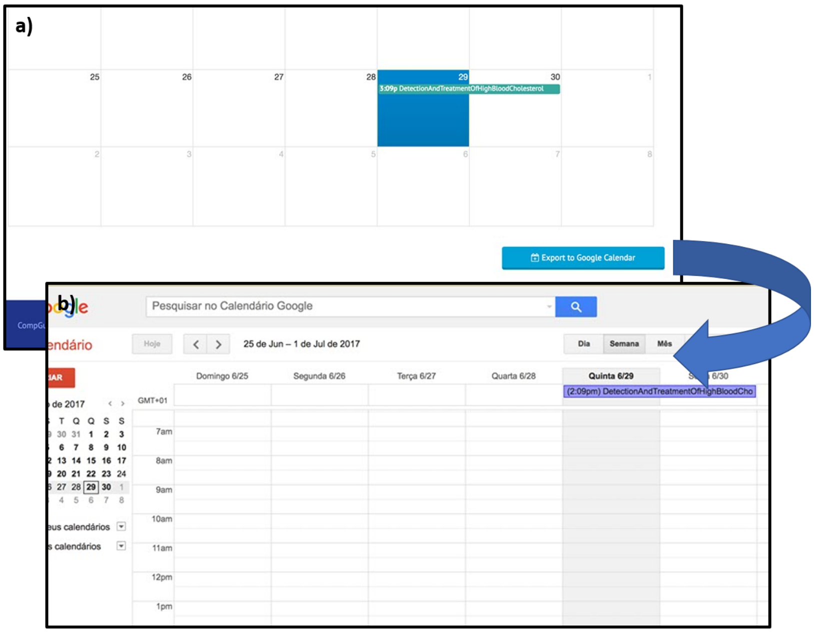

a)
3:09p DetectionAndTreatmentOfHighBloodCholesterol
Export to Google Calendar
b)
Pesquisar no Calendário Google
25 de Jun – 1 de Jul de 2017
Hoje
Dia
Semana
Mês
Domingo 6/25
Segunda 6/26
Terça 6/27
Quarta 6/28
Quinta 6/29
(2:09pm) DetectionAndTreatmentOfHighBloodCho
GMT+01
7am
8am
9am
10am
11am
12pm
1pm

Fig. 7: Calendar view of clinical procedures in Google calendar.

5 Execution Examples

To test our temporal ontology, we used the NCCN Clinical Practice Guideline for Colon Cancer [25]. Its representation resulted in an OWL file containing 223 task instances, of which: 190 were Action tasks, 21 were Question tasks, one was a Decision task, and 11 were Plans. Out of the 223 tasks, a total of 95 had temporal constraints. The representation of the NCCN guideline in the model was carried out using Protégé, an ontology editor for OWL. 4.1. This CPG includes procedures that unfold over different phases of treatment, from cancer staging to follow-up, and presents a wide variety of temporal patterns. The most abundant pattern was the *Periodicity*, mainly because of the rich description of chemotherapy regimens made in this protocol.

As demonstrated in [20], the temporal ontology was able to represent effectively all the temporal patterns in the CPG, with a special focus on *Durations* and *Periodicities*, since they were the most frequent temporal aspects. Considering an example of a task in the form of a clinical *Action* from the CPG, which we will refer to as Example 1 from now on, the use of a *Duration* constructor may be derived from the following description ”perform neoadjuvant therapy for six months”. In it, the *Action* consists in neoadjuvant therapy (a term used to refer to chemotherapy or radiotherapy) before treatment with a *Duration* expressed using an exact duration value of six and a temporal unit of month.

Regarding periodic tasks, most of them were also bounded by a *Duration*. The constraints followed a structure similar to the one in the recommendation "apply medication for neoadjuvant therapy every two weeks for two-three months", which we will consider as Example 2. It is possible to identify the *Action* to apply medication for neoadjuvant therapy, the periodicity value of two with a temporal unit of week, a minimum duration value of two, a maximum duration value of three, and the respective temporal unit of month. In this case, the execution engine would recommend the execution of the task with the specified frequency at least for two months and at most for three.

The *Guideline Execution Engine* from the CompGuide architecture is used to produce inferences that ultimately result in recommendations of clinical tasks. Once these recommendations are retrieved, their constraints (in this case, their temporal constraints) are interpreted by the Personal Assistant Web Application and mapped onto the different views mentioned earlier. With this, for Example 1, an event with a duration of 6 months is created, starting on the 18th of July of 2017, as shown in Figure 8 a), and finishing on the 16th of January of 2018, as can be seen in Figure 8 c). The corresponding result for the expression that concerns Example 2 consists of a set of events that repeat every two weeks, so the application will unfold the recommendation in multiple events and register them in the timeline. Although the execution engine would recommend the execution of the task with the frequency at least for two months and at most for three, the *Personal Assistant Web Application* will display the maximum duration (three months) because it is the upper bound of the task. Nonetheless, the task controllers will notify the health care professional when the minimum duration is achieved. As such, the result would be six new calendar events from the start date of the task execution up to three months. The first and second events start on the 18th of July and 1st of August, as shown in the Figure 9 a). The third and fourth start on the 15th and 29th of August, as depicted in Figure 9 b). Finally, the fifth and sixth events start on 12th and 26th of September, as shown in Figure 9 c). Then, the user can consult on the timeline and calendar widgets the scheduling of these events in order to execute the clinical task and manage its completion. Whenever the users should execute the tasks or when they should start them, the application provides notifications, as side messages, about the different temporal constraints, thus alerting the user.

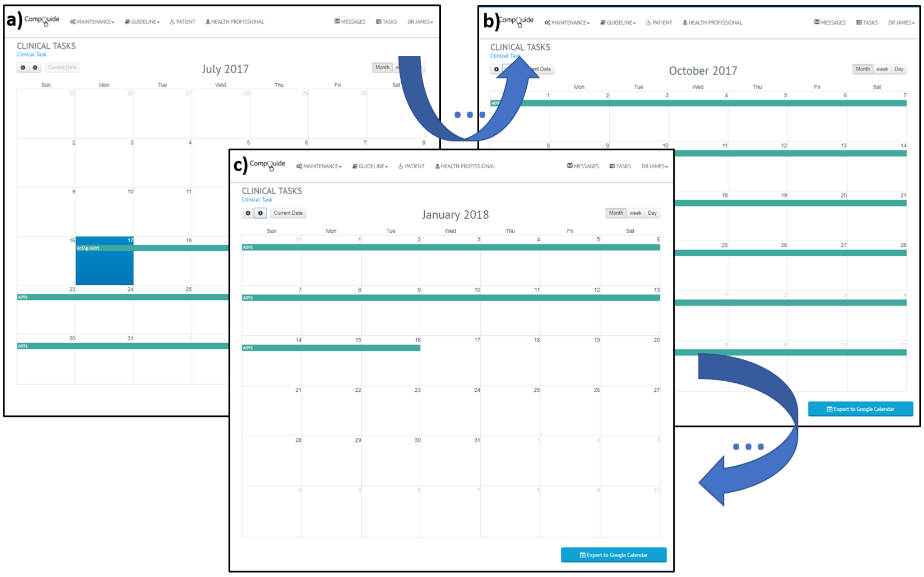


Fig. 8: Execution of a clinical task from Example 1, as can be seen in the Personal Assistant Web Application. Figures a), b), and c) show different consecutive execution times.

6 Conclusions and Future Work

The CompGuide system presented herein aims to increase the reach of CIGs beyond the medical office. The purpose of the different implementations is to ensure the timely enactment of clinical procedures over the course of patient management, removing the possibility of inadvertently skipping steps that may prove to be crucial later on for his recovery. In addition to decision support functionalities, common to other CIG systems, the CompGuide system allows the development of additional scheduling and alert features to assist the health care professional in keeping track of their patients. Therefore, its main contribution is a new method to integrate CPG advice in a clinical setting and make it easily available. The *Guideline Execution Engine* included in the *Core Server* establishes the relative order of tasks to be executed and their execution times based on the clinical information retrieved from the patient. This is the most complex part of CIG deployment, given the complexity, the procedural and temporal patterns of CPGs may show. Once these constraints are produced and delivered through a distribution model, in the form of guideline as a service, it becomes possible to develop reminder tools like the ones described herein. Here lies a development that can close the gap between CPGs and practitioners and promote the integration of evidence-based clinical advice in AAL monitoring systems.

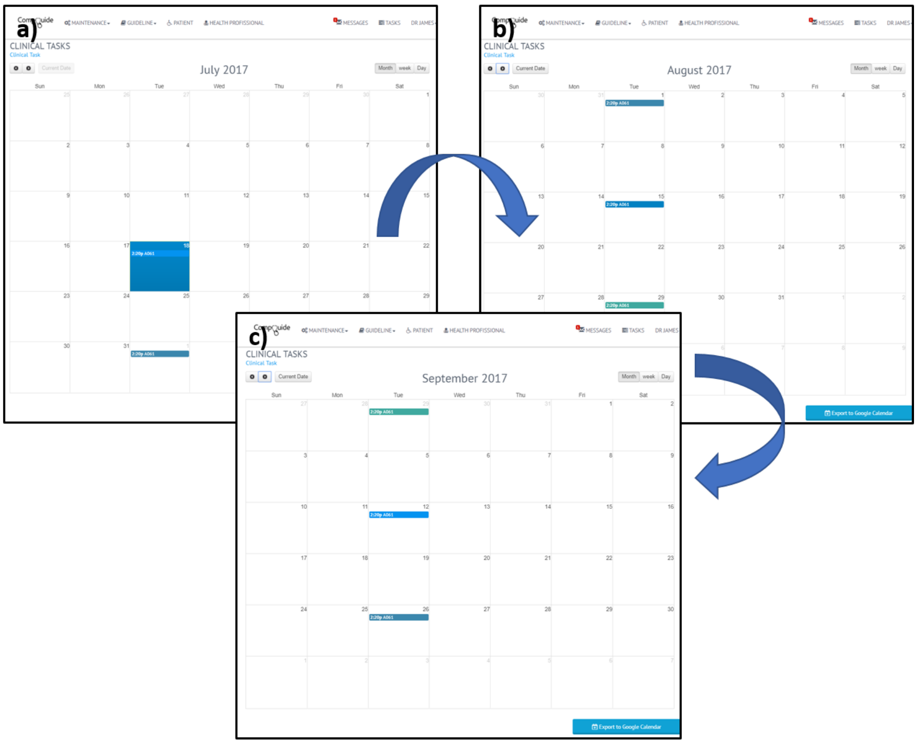


Fig. 9: Execution of a clinical task from Example 2, as seen in the Personal Assistant Web Application. Figures a), b), and c) show different consecutive execution times.

This mapping of the clinical tasks onto a temporal execution line raises a relevant question. The *modus operandi* of the Personal Assistant Web Application is to issue notifications and alerts in order to promote compliance from the physician. However, if tasks are not executed at their appropriate times, the tool only issues alerts and allows the physician to skip the task and move to the next one. There are other methods to manage this situation, but all of them have drawbacks. Re-scheduling the task may imply verifying if the state of the patient allows the enactment of the procedure at a later time. Not performing the task may be equally damaging to the patient. Such an issue will be under consideration in future developments of the system. Additionally, we recognise the need for an evaluation of the system and both the Personal Assistant Web Application and the health care assistant mobile app. Such can be done by through an experiment in which a physician uses the system and its two implementations to obtain advice about the patients he is responsible for. In addition to usability assessments, with this experiment, it will be possible to compare the recommendations provided by the system to those the health care professional would usually issue. It is our intention to conduct this study and obtain an assessment of the fitness of the system to CIG deployment.

Acknowledgements

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.

References

1. S. Silberstein, Journal of Neurosurgery Pediatrics **25**(10), 765 (2005). DOI 10.1111/j.1468-2982.2005.01014.x
2. K.N. Lohr, M.J. Field, et al., *Clinical practice guidelines: directions for a new program*, vol. 90 (National Academies Press, 1990)
3. M. Peleg, Journal of Biomedical Informatics **46**(4), 744 (2013). DOI 10.1016/j.jbi.2013.06.009
4. D. Wang, M. Peleg, S.W. Tu, A.A. Boxwala, O. Ogunyemi, Q. Zeng, R.A. Greenes, V.L. Patel, E.H. Shortliffe, Journal of Biomedical Informatics **37**(5), 305 (2004). DOI 10.1016/j.jbi.2004.06.002
5. O. Young, Y. Shahar, Artificial Intelligence Review **3581**(1), 166 (2005)
6. P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G. Molino, G. Correndo, Studies in Health Technology and Informatics **101**(3), 162 (2004). DOI 10.3233/978-1-60750-944-8-162
7. D. Berg, P. Ram, J. Glasgow, J. Castro, The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society **4**, 3217 (2004). DOI 10.1109/IEMBS.2004.1403906
8. M.A. Musen, Y. Shahar, E.H. Shortliffe, in *Biomedical Informatics*, ed. by E. Shortliffe, J. Cimino, Health Informatics (Springer New York, 2006), pp. 698–736. DOI 10.1007/0-387-36278-920
9. K. Kaiser, S. Miksch, Artificial intelligence in medicine **46**(1), 55 (2009). DOI 10.1016/j.artmed.2008.08.009
10. T. Oliveira, P. Novais, J. Neves, in *Trends in Practical Applications of Agents and Multiagent Systems SE - 10*, *Advances in Intelligent Systems and Computing*, vol. 221 (Springer International Publishing, 2013), pp. 77–85
11. A. Silva, T. Oliveira, J. Neves, P. Novais, *Transforming Medical Advice into Clinical Activities for Patient Follow-Up* (Springer International Publishing, Cham, 2017), pp. 169–176. DOI 10.1007/978-3-319-60285-1_14
12. D. Isern, A. Moreno, International Journal of Medical Informatics **77**(12), 787 (2008). DOI 10.1016/j.ijmedinf.2008.05.010
13. A.a. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q.T. Zeng, D. Wang, V.L. Patel, R.a. Greenes, E.H. Shortliffe, Journal of Biomedical Informatics **37**(3), 147 (2004). DOI 10.1016/j.jbi.2004.04.002
14. Y. Shahar, S. Miksch, P. Johnson, Artificial intelligence in Medicine **14**(1-2), 29 (1998). DOI 10.1016/S0933-3657(98)00015-3
15. P. Novais, R. Costa, D. Carneiro, J. Neves, J. Ambient Intell. Smart Environ. **2**(2), 179 (2010)
16. L. Lima, P. Novais, J. Neves, C.J. Bulas, R. Costa, Logic Journal of the IGPL **19**(2), 315 (2011). DOI 10.1093/jigpal/jzq029
17. A. Costa, P. Novais, J.M. Corchado, J. Neves, Logic Journal of the IGPL **20**(4), 689 (2012)
18. R. Picking, A. Robinet, V. Grout, J. McGinn, A. Roy, S. Ellis, D. Oram, The Computer Journal **53**(6), 842 (2010). DOI 10.1093/comjnl/bxp089

19. J. Jiang, F. Khelifi, P. Trundle, A. Geven, Journal of Assistive Technologies **3**(3), 27 (2009). DOI 10.1108/17549450200900022
20. T. Oliveira, A. Silva, J. Neves, P. Novais, Journal of Medical Systems **41**(1), 1 (2017). DOI 10.1007/s10916-016-0655-6
21. T. Oliveira, P. Leão, P. Novais, J. Neves, in *Trends in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection SE - 18*, *Advances in Intelligent Systems and Computing*, vol. 293, ed. by J. Bajo Perez, J.M. Corchado Rodríguez, P. Mathieu, A. Campbell, A. Ortega, E. Adam, E.M. Navarro, S. Ahrndt, M.N. Moreno, V. Julián (Springer International Publishing, 2014), pp. 149–156. DOI 10.1007/978-3-319-07476-4{_}18
22. E. Shalom, Y. Shahar, E. Lunenfeld, Journal of Biomedical Informatics (November) (2015). DOI 10.1016/j.jbi.2015.11.006
23. Riontech/customcalendar. Custom Android Calendar. URL `https://github.com/Riontech/CustomCalendar`
24. G. Developers. Get Started with the Calendar API (2016). URL `https://developers.google.com/google-apps/calendar/overview`
25. A. Benson, T. Bekaii-Saab, E. Chan, Y.J. Chen, M. Choti, H. Cooper, P. Engstrom, NCCN Clinical Practice Guideline in Oncology Colon Cancer. Tech. rep., National Comprehensive Cancer Network (2013)

3.3 Enhancing Decision Making by Providing a Unified System for CIG Management

Title	Enhancing Decision Making By Providing A Unified System For CIG Management
Authors	António Silva, Tiago Oliveira, Filipe Gonçalves, and Paulo Novais
Publication Type	Journal
Publication Name	Expert Systems
Publisher	Wiley Online Library
Volume	38
Year	2019
Month	May
DOI	https://doi.org/10.1111/exsy.12412
State	Published online
Scimago journal rank (2019)	0.348, Artificial Intelligence (Q3)

Contribution of the doctoral candidate
The doctoral candidate, António José Linhares da Silva, declares to be the main author and the major contributor of the paper *Enhancing Decision Making By Providing A Unified System For CIG Management*.

Expert Systems
DOI 10.1111/exsy.12412

Enhancing Decision Making By Providing A Unified System For CIG Management

António Silva, Tiago Oliveira, Filipe Gonçalves, Paulo Novais

Received: 5 September 2018 / Accepted: 15 March 2019

Abstract The need for integration of Clinical Practice Guidelines (CPGs) in daily clinical practice calls for computational systems able to operationalise their knowledge and provide an enhanced experience in their enactment. Current approaches lack in functionalities such as scheduling and temporal management of CPGs, the combination of CPGs and user-friendly systems for Computer-interpretable guideline creation and editing. This paper presents a comprehensive architecture for the deployment of Computer-interpretable Guidelines (CIGs), featuring components that allow: the creation and manipulation of clinical practice guideline knowledge elements, execution of CIGs with the temporal verification of clinical tasks, and drug conflict identification and resolution. This comprehensive approach provides a step-by-step assistant for healthcare professionals in the form of an agenda of activities that detects drug interactions when they are prescribed simultaneously and applies a mitigation algorithm to select possible and conflict-free alternatives. This work addresses the lack of a unified pipeline and mitigation features shown in approaches to CIG conflict mitigation in use.

Keywords Computer-Intepretable Guidelines · Clinical Decision Support · Ontologies · Drug-drug Interactions · Conflict Resolution.

A. Silva · Filipe Gonçalves
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: {asilva, fgoncalves}@algoritmi.uminho.pt

T. Oliveira
National Institute of Informatics, Tokyo, Japan
E-mail: toliveira@nii.ac.jp

P. Novais
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: pjon@di.uminho.pt

1 Introduction

To lead to a better acceptance of clinical practice guidelines (CPGs) by healthcare professionals at the point of care, there have been several efforts to computerise CPGs in Computer-interpretable Guidelines (CIGs) and incorporate them within Clinical Decision Support Systems (CDSSs). CIGs are representations of CPGs in a structured and machine-readable digital format. They can be integrated into CDSSs in order to facilitate the implementation of good recommendations in daily clinical practice since they are available during the clinical act [1]. The integration of CPGs in CDSSs can lead to better acceptance and application of these good medical practices because these systems are able to monitor the actions and observations of health professionals [2] [3].

In computer science, several languages and tools exist to help users and system developers to create good and effective CIGs. Some models aim at representing and executing CPGs as CIGS; others specialise in CIG acquisition and representation.

Although current solutions offer tools to create and execute CPGs, they lack in functionalities such as scheduling and temporal management of CPGs, the combination of CPGs and user-friendly systems for Computer-interpretable guidelines creation and editing. Others, despite being filled with many functionalities and tools, are difficult to use, with complicated and non-intuitive interfaces making them less suitable for daily use in such a complicated environment as health care delivery. These models also lack the flexibility to support cases where multiple protocols need to be combined, which can be problematic in the case of *multimorbidity.* This refers to the condition of a patient who suffers from multiple health conditions and for whom a consistent treatment has to be devised so as to avoid potential drug-drug and drug-disease interactions.

The work described herein presents a unified tool that aims to represent CPGs as a CIG through the CompGuide Editor and provides a system that promotes better integration of CPG recommendations in the daily life of healthcare professionals by producing an agenda containing clinical recommendations, allowing the execution of CIGs [4,5]. The system also allows identifying recommendation interactions, conflicts, and alternatives using existing terminology services, namely, the RxNorm API [6]. The present work uses the CompGuide ontology [7], a model which provides a representation of CPGs and the CompGuide Editor, a plugin that permits the management of CIGs by allowing their creation and editing. By integrating these two components in a unified workflow, we intend to propose an architecture for CIG acquisition, editing and execution. Moreover, we will also provide details on how we address some of the identified limitations of the current solutions for CIG execution and representation, particularly those that concern the identification of drug conflicts. Thus, the contributions emphasised in this work are the complete integration of the stages in a CIG development life cycle, direct expedience of CIGs to a clinical situation, a unified workflow for the acquisition and customisation of CIGs, automatic identification of conflicts between clinical recommendations and provision of alternative recommendations that address these conflicts. The present paper represents an extension of the work in [8].

Section 2 describes related work regarding systems for combining CPGs. In Section 3, we present a CompGuide architecture proposal for CIG management and execution as well as the contributions for the deployment of CPGs in CDSSs. Section 4 describes the functionalities supporting care with examples of CIG execution. Finally, Section 5 presents conclusions about the work developed so far and future work considerations.

2 Existing Approaches to Combining CPGs - Multimorbidity

There are two purposes for CIG description languages: support the acquisition and editing of CPGs, either manually or semi-automatically, and the modelling of the workflow structure of CPGs for CIG execution. Regarding these purposes, several approaches have been developed. Some approaches focus in CIG execution, such as Arden Syntax [9], Guideline Interchange Format (GLIF) [10], Asbru [11], PROforma [12] and the Standards-based Active Guideline Environment (SAGE) [13]. Others aim at representing and acquiring the knowledge from CPGs in CIGs. Examples include Protégé [14], SAGE Workbench [15], Tallis [16], GEM Cutter [17] and Asbru View [18].

Although current solutions offer ways to represent, acquire and execute CPGs, they lack in the verification of conflicts and interactions resulting from the execution of concurrent guidelines. Several conflicts and interactions can happen when applying concurrently multiple CPGs to patients, including adverse drug events, increased treatment complexity, the burden of disease and cost of treatment [19] [20]. Thus, the application of multiple CPGs individually can result in complex multiple drug regimens (polypharmacy) with the potential for harmful combinations of drugs [21].

Current solutions also lack mechanisms for integrating and scheduling CIG recommendations in the daily routine of health care professionals. These mechanisms include patient tracking, patient follow-up, scheduling of procedures, and the monitoring of procedure constraints, and, at the same time, autonomously inform healthcare professionals about essential aspects of these procedures in the most diverse situations.

In the following sections, we describe different systems that automatically identify the possible interactions between concurrent CPGs for multimorbid patients.

2.1 Constraint Logic Programming

Wilk et al. propose a model that follows the constraint logic programming mathematical paradigm and combines it with constraint satisfaction problems [22]. They use CIGs as an activity graph and use constraint logic programming to identify and mitigate possible adverse interactions between CIGs, identifying conflicts associated with potentially contradictory and adverse activities that happen when applying multiple guidelines on the same patient. The objective is to alert healthcare professionals about the possible conflicts during the definition of the treatment plans [23]. Although this approach provides automatic identification of conflicts and solutions, it depends on the availability of knowledge bases containing information about both diseases and manual labour for combining CIGs. So, in order to provide automatic identification and resolution of conflicts, solutions need to be defined in a *medical background knowledge* as protocol-dependent rules/constraints. In their approach, Wilk et al. do not consider temporal aspects of CPGs.

2.2 Rule-based Combinations

The rule-based combination methodology has been developed for identification and reconciliation of drug conflicts between recommendations of two concurrently executed CPGs [24]. They use a standard terminology called ATC (Anatomical Therapeutic Chemical Classification System for drugs) in order to provide as output, a final treatment plan without interaction comprising a set of ATC-codes of medicines that should be prescribed.

For the identification of all possible drug conflicts that can occur when combining two specific CPGs, they use the knowledge from health care professionals and knowledge engineers in order to manually build knowledge units for the pairwise combination of three diseases: Hypertension, Diabetes Mellitus and Heart Failure. These Knowledge units rely on the existence of drug-drug interactions, the presence of a drug which is adverse to a specific disease (drug-disease interaction) and the absence of a necessary drug for a combination of diseases.

Although this approach can only combine CPGs pairwise, a final treatment plan based on two CPGs could again be combined pairwise with a new CPG.

2.3 OntoMorph

Jafarpour et al. approach uses a collection of ontologies to represent the guidelines, the general domain, the mappings between guidelines and the general domain, and decision rules for simultaneous execution of guidelines that are provided by domain experts [25]. They used ontologies to develop systems to merge two concurrent CPGs into a co-morbid personalised guideline. They extracted clinical tasks from the CPG and converted them to computer-interpretable rules with an OWL-based CPG ontology. An ontology is a methodology for CPG representation. It consists of rules to represent declarative knowledge (medical statements and propositions) and procedural knowledge (workflow structures and actions). OWL is a W3C standard for web ontologies, for which CPG concepts are converted to RDF triplets and an XML file [26]. This model defines four types of constraints for concurrent execution of tasks from multiple guidelines: *workflow constraints*, *operational constraints*, *temporal constraints* and *medical constraints*.

Workflow constraints are rules that specify whether tasks should be combined with, substituted by, executed simultaneously with or executed before or after a task from another guideline. *Operational constraints* refer to limitations for combining tasks at a specific medical institute; *temporal constraints* specify the time required between the first and second task of two guidelines. *Medical constraints* are divided into task substitutes (a substitute for a task of protocol A that does not conflict with a task of protocol B) and use results constraints (a rule that specifies expiry date of task results).

The work also includes a merging representation ontology to capture merging criteria in order to achieve the combination of CIGs. Semantic Web Rule Language (SWRL) rules were used to identify potential conflicts during the merging process. All conditions related to the merging process need to be described by the rules, increasing the effort to maintain the system up-to-date, and reducing the possibility of sharing knowledge. However, some related limitations were not yet entirely addressed in their work, for instance, potential contradictions between rules, the scalability of the merging model to combine several CIGs, and how the ontology/rules are maintained up-to-date.

2.4 Transition-based Medical Recommendations Model

The TMR4I model has been developed for the automatic inference of interactions between recommendations [27]. Its scope is currently limited to conflicts between CPG statements on drug prescription, but it could be used for non-pharmacological treatment recommendations as well.

This model defines meta-rules for identification and reconciliation of three categories of drug conflicts using SPARQL queries (SPARQL is a W3C-standard for semantic queries). The meta-rules define how a conflict is identified, and how drugs with similar effects but without conflicts can be selected from CPG-knowledge. The categories of conflicts within CPGs are *repetition interactions*, *contradiction interactions* and *alternative interactions*.

A web-tool for execution of guidelines was developed. In this tool, clinicians firstly enter all guideline recommendations applicable to a patient. The execution engine creates a new, merged guideline with all recommendations. With the SPARQL meta-rules, interactions are identified and classified. Then, the engine consults the alternative recommendations, in order to choose solutions for the conflicts. Finally, a list of conflicts and recommended solutions is presented to the clinician.

3 An Architecture for CIG Management with CIG Interaction Detection and Resolution

The work described herein presents a unified workflow for CIG management that encompasses the phases of CIG representation, acquisition and execution. This work uses the CompGuide plugin editor [28] and CompGuide CIG execution server, providing a unified architecture that aims to reuse and integrate knowledge elements from these two components. Moreover, we provide a system that automatically identifies recommendation interactions (more specifically drug interactions), conflicts, and alternatives using existing terminology services such as the RxNorm API [6]. The goal of RxNorm is to link different drug nomenclatures and exchange data efficiently between them and can be used to check interactions between drugs. The specified architecture is shown in Figure 1. The following sections provide an explanation of the architecture in the different stages regarding the deployment of CIGs in the system.

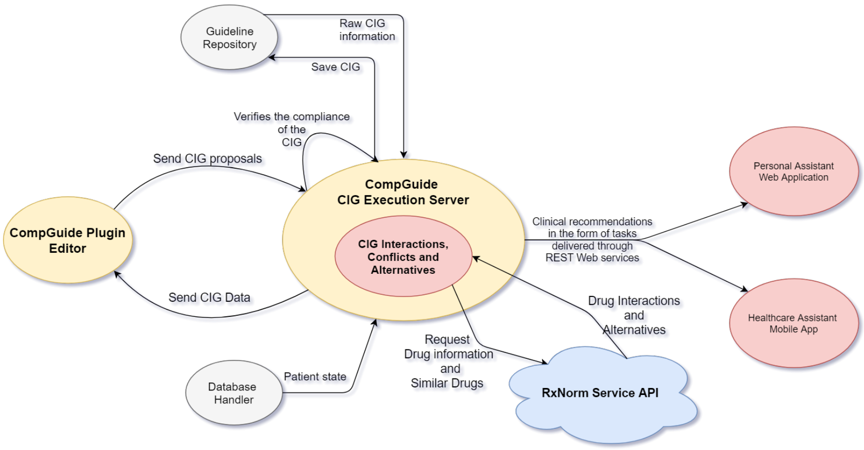
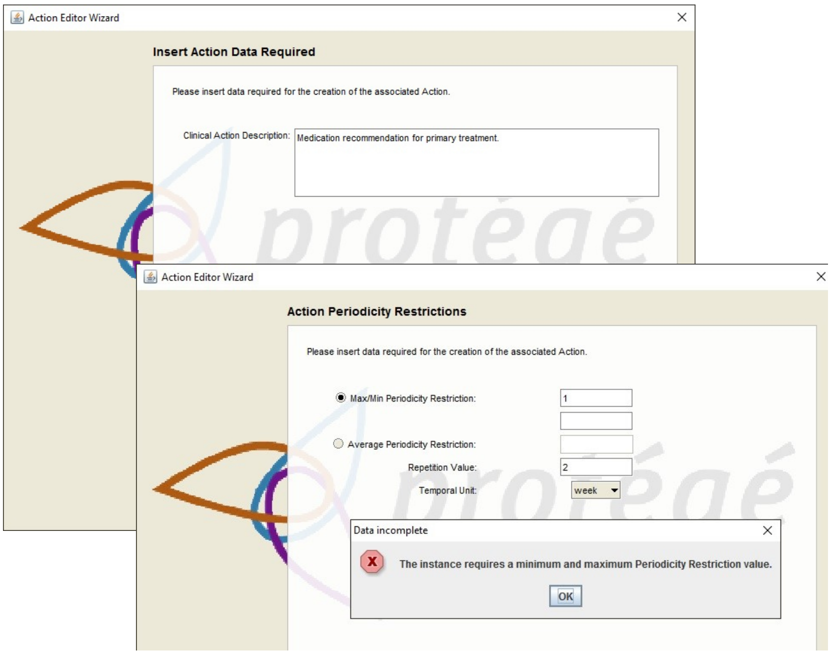

Guideline Repository
Raw CIG information
Save CIG
Verifies the compliance of the CIG
Send CIG proposals
CompGuide Plugin Editor
CompGuide CIG Execution Server
CIG Interactions, Conflicts and Alternatives
Send CIG Data
Personal Assistant Web Application
Clinical recommendations in the form of tasks delivered through REST Web services
Healthcare Assistant Mobile App
Drug Interactions and Alternatives
Request Drug information and Similar Drugs
RxNorm Service API
Database Handler
Patient state

Fig. 1 Architecture for the management of CIGs.

3.1 CIG Representation and Acquisition

The present work uses the CompGuide ontology to represent CPGs in the form of a task network. The CompGuide ontology [29] contains different types of clinical tasks and provides a model of temporal representation [30] that aims to represent the temporal constraints placed on clinical tasks. To acquire and represent CPGs we use the CompGuide Plugin Editor which provides information step-by-step how to fill the data for the guideline entries [28]. This plugin performs the role of managing the creation and editing of CIGs.

The CompGuide Plugin Editor is a *Protégé Desktop* plug-in interface that allows the creation and editing of CIGs in a quick and simple way. Its design as a plugin allows the use of all the features offered by the *Protégé Desktop* application, namely the functionality of managing the data of an ontology through the use of a graphical interface, along with the creation of new features. Also, this plugin provides features such as the capacity to re-utilise the knowledge units of the CompGuide ontology as well as the capability to verify if the required data is correctly inserted while proceeding with the creation/editing process, as depicted in Figure 2.



Action Editor Wizard

Insert Action Data Required

Please insert data required for the creation of the associated Action.

Clinical Action Description: Medication recommendation for primary treatment.

Action Editor Wizard

Action Periodicity Restrictions

Please insert data required for the creation of the associated Action.

● Max/Min Periodicity Restriction: 1

○ Average Periodicity Restriction:

Repetition Value: 2

Temporal Unit: week

Data incomplete

The instance requires a minimum and maximum Periodicity Restriction value.

Fig. 2 Failed verification based on incomplete inserted data.

These views include: the OntoGraf View that allows the graphical representation of the ontology; the Individuals by Type View, which shows all the individuals (sorted by respective OWL classes) saved in the ontology; and CompGuide Wizard Options View, which has the set of features to manage the individuals and their relations, plus download/upload the CompGuide ontology file. This view also shows the total number of individuals saved in the loaded ontology. The use of this plugin allows the management of CIGs, providing access to features, such as creation, editing and deletion of data concerning these structured representations.

3.2 CIG execution

After the creation and editing of CIGs, the proposals will be sent to the *CIG Execution Server*, which is responsible for making the consistency check and the verification of compliance of the CIGs, according to the CompGuide ontology. The *CIG Execution Server* saves the documents in the *Guideline Repository*, which keeps different CIGs, organised by authorship and by date. Furthermore, the *CIG Execution Server* is responsible for providing CIG data to the *CompGuide plugin editor*, in order to ensure that the last version of the guideline is being edited.

The *CompGuide CIG Execution Server* accesses the information about the patient state provided by the *Database Handler* as well as temporal constraints on the execution of the clinical tasks and temporal constraints on the state of a patient given by the *Guideline Repository* to produce recommendations. Thus, this component interprets all the scheduling constraints on the tasks and produces enactment times. For each produced recommendation, the *CompGuide CIG Execution Server* is responsible for retrieving drug information and for each pair of drugs calls the RxNorm Interaction API to obtain the severity and description of the interaction [6].

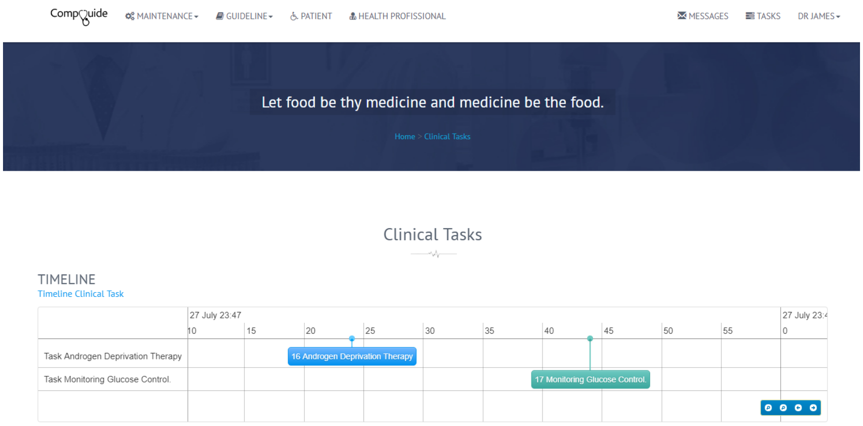
With information about the severity of the interaction, it is possible to determine if alternative drugs are needed. The severity can assume two values: *N/A*, if it is not possible to qualify the severity of the interaction and *high* if an adverse interaction exists. In case that an interaction exists, the CIG Execution Server is responsible for calling the RxNorm RxClass API, which allows obtaining alternative drugs [6]. This service provides information such as similarity scoring (a score that determines the similarity between drugs), the drug name, the source of the drug relations and the relationship of the drug class to its members. Later, the applications developed to interact with healthcare professionals are then responsible for checking the start and end dates of clinical tasks. This aspect is further explained with a case example in Section 4.

3.3 Visualisation of Clinical Recommendations

The *Personal Assistant Web Application* and *Healthcare Assistant Mobile App* access the data through the web services available in the CompGuide system [31]. The *Personal Assistant Web Application* is a web application developed in Java Server Faces (JSF), which follows the Model-View-Control (MVC) paradigm. The *Health Care Assistant Mobile Application* is an android application developed in Java.

These two components aim to allow the healthcare professional to monitor and follow-up of the clinical process of the patients, providing and displaying clinical recommendations in different forms.

The *Personal Assistant Web Application* provides a timeline view, in which all the clinical tasks are shown in a schedule. The activity timeline allows compressing multiple tasks into a single continuity without compromising succession and easy understanding of clinical recommendations. In addition, representing clinical recommendations on a timeline brings benefits, such as the ability to sequence events and reduce the potential to overwhelm health professionals. This view is shown in Figure 3.



Let food be thy medicine and medicine be the food.

Clinical Tasks

TIMELINE
Timeline Clinical Task

Task Androgen Deprivation Therapy — 16 Androgen Deprivation Therapy

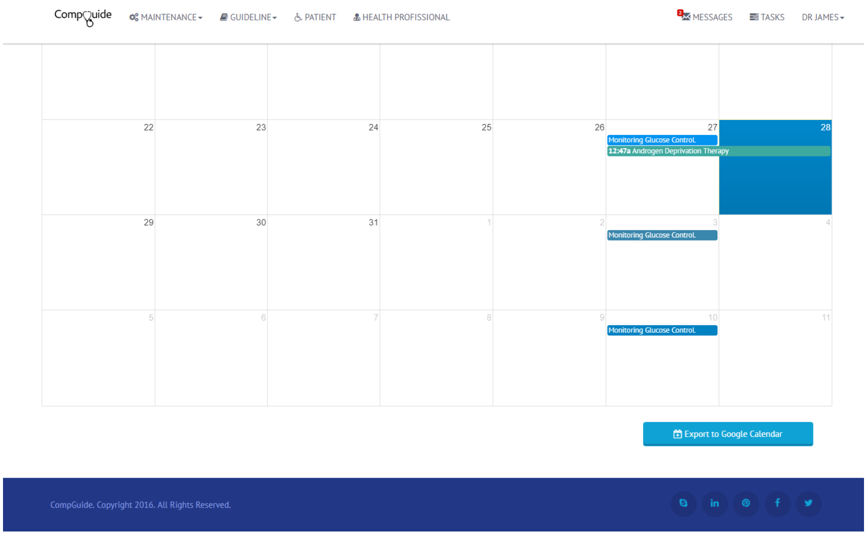
Task Monitoring Glucose Control. — 17 Monitoring Glucose Control.

Fig. 3 Timeline view of clinical procedures in the CompGuide Personal Assistant Web Application.

The *Personal Assistant Web Application* provides another view, a calendar in which the healthcare professional can visualise the tasks according to the temporal granularity. The temporal granularity can be represented by day, week or month. While in the timeline view it is more comfortable to detect the start and end dates of tasks, with calendar view it is easier to understand the temporal constraints that link clinical tasks, the execution sequences, the duration of the tasks and waiting times between recommendations. Figure 4 shows the same tasks as in the timeline but displayed over a week, where it is possible to verify, for instance, for how long a clinical task should be applied.

The *Health Care Assistant Mobile App* features a calendar widget that allows viewing the clinical tasks and schedule new tasks. The *Health Care Assistant Mobile App* presents a calendar widget which provides the view and methods necessary to display a calendar and schedule events. This calendar allows navigation through the chosen temporal granularity (day, week and month), as well as allowing to view the details of the events, as depicted in Figure 5.

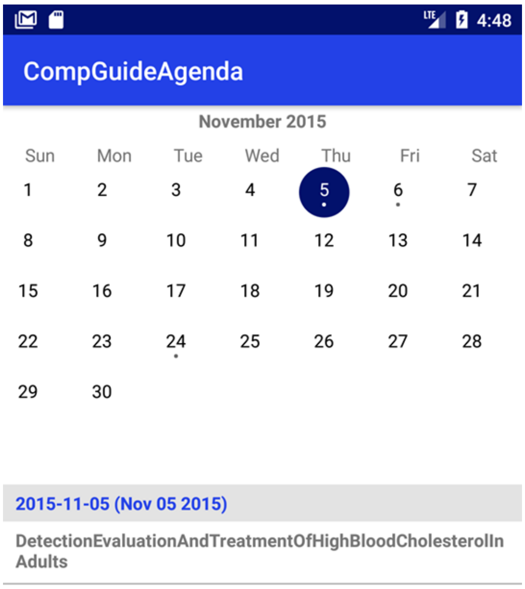
In order to ensure the execution of tasks at the designated time, a notification system supported by a message box was implemented. These elements are both shown in Figure 6. The message box features messages such as indications about the tasks that should be performed or should have already been performed, offering the possibility to mark them as executed. As for the notification system, it is used to periodically alert the user about task enact-



22	23	24	25	26	27 Monitoring Glucose Control. 12:47a Androgen Deprivation Therapy	28
29	30	31	1	2	3 Monitoring Glucose Control.	4
5	6	7	8	9	10 Monitoring Glucose Control.	11

CompGuide. Copyright 2016. All Rights Reserved.

Fig. 4 Calendar view of clinical procedures in the CompGuide Personal Assistant Web Application.



November 2015

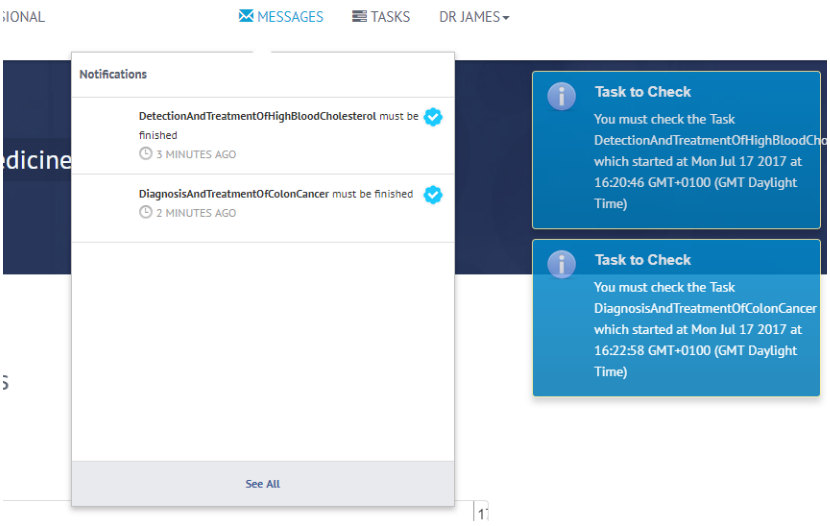
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

2015-11-05 (Nov 05 2015)

DetectionEvaluationAndTreatmentOfHighBloodCholesterolIn Adults

Fig. 5 Calendar view of clinical procedures in the mobile application.

ment times and steps to collect information about the patient, such as the outcomes of clinical tasks, according to their respective temporal restrictions. The notifications are shown as pop-up messages.



DetectionAndTreatmentOfHighBloodCholesterol must be finished
3 MINUTES AGO

DiagnosisAndTreatmentOfColonCancer must be finished
2 MINUTES AGO

Task to Check
You must check the Task DetectionAndTreatmentOfHighBloodCho which started at Mon Jul 17 2017 at 16:20:46 GMT+0100 (GMT Daylight Time)

Task to Check
You must check the Task DiagnosisAndTreatmentOfColonCancer which started at Mon Jul 17 2017 at 16:22:58 GMT+0100 (GMT Daylight Time)

Fig. 6 Message box and notification in the CompGuide Personal Assistant Web Application.

4 Case Example of Execution and Conflict Mitigation

To exemplify the capabilities of the system, we will show how recommendations from different CIGs are executed, visualised and analysed, using two CIGs based on the NCCN Clinical Practice Guideline for Prostate Cancer [32] and the IDF Clinical Practice Recommendations for managing Type 2 Diabetes [33]. These guidelines were an excellent case study since they provided several types of tasks, various temporal constraints and several conflicts among the guidelines. Thus, the representation of these guidelines in the CompGuide model allowed the identification of different temporal constraints as well as the drug interactions from recommended actions in the CompGuide ontology. The CompGuide plugin editor supported all the work of creating and editing the instances in the CompGuide ontology.

For demonstrative purposes, we will consider two recommendations. Recommendation A is ”Apply insulin 0.2 units/kg and titrate once weekly at one unit each time during six months to achieve a target fasting blood glucose between 3.9 and 7.2 mmol/L (70 and 130 mg/dL)” from the Type 2 Diabetes CIG. Recommendation B is: ”Apply goserelin, leuprolide, histrelin 180 mg/m2 or Triptorelin 100mg/m2 as part of Androgen Deprivation Therapy” extracted from the prostate cancer CIG.

Temporal constraints in CompGuide are defined through the use of values and respective temporal units to allow the specification of waiting times, durations, and periodicities. Through the analysis of the expression of recommendation A, it is possible to identify the action to apply insulin, a periodicity value of 1 with a temporal unit of week, a duration value of six, and the respective temporal unit of month. In this case, starting on the 18th of July of 2018 the system will create one event for each week with a duration of one day, during 6 months. Therefore, the expected conclusion of this task will be on the 18th of January of 2019, as can be seen in Figure 7. As for recommendation B, the action to apply goserelin, leuprolide, histrelin or triptorelin can be identified, with a duration value of 1 and temporal unit of day, starting and finishing on the 18th of July of 2018.

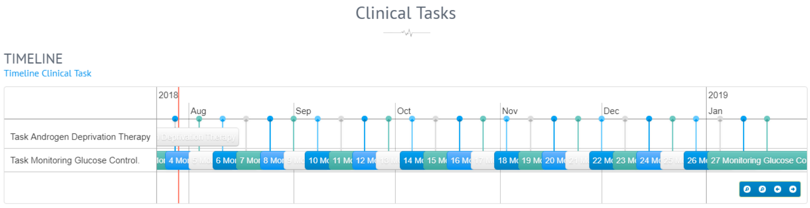
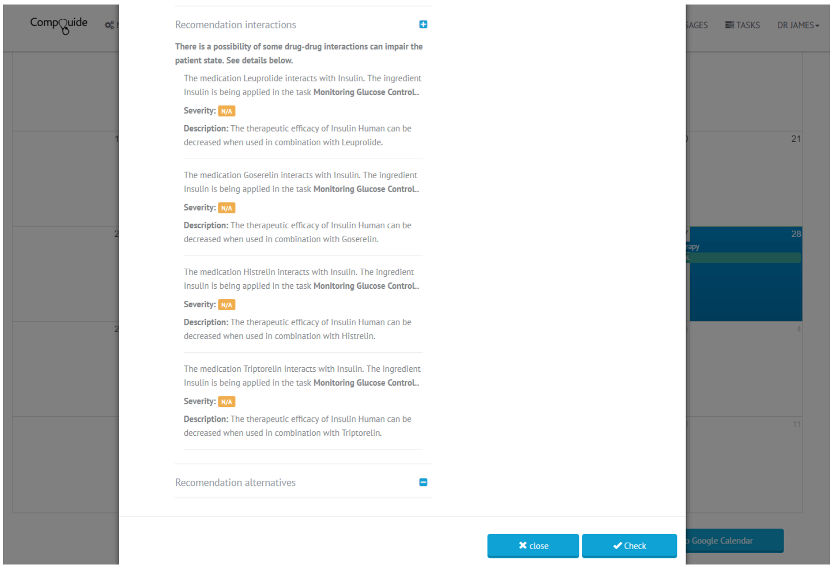


Fig. 7 Execution of a clinical task from the given expression, as seen in the Personal Assistant Web Application.

In this particular case, two guidelines are concurrently being executed. recommendation A interacts with recommendation B since the medications leuprolide, gserelin, histrelin and triptorelin applied in recommendation B have adverse effects on the therapeutic efficacy of insulin. In this case, the application will display these conflicts as shown in Figure 8, as identified through the use of the RxNorm API.

With the severity of the interactions and conflicts between recommendations, the system tries to provide alternative recommendations. For this purpose, the system provides a set of alternative drugs. For the set of alternatives, the system calculates which one will be applied through a mitigation function to determine which alternatives best fit the needs of users. This function may have different mitigation principles, such as the similarity between drugs or user preferences. One possible principle is a multiple criteria mechanism for supporting decision making such as Multiple-criteria Decision Analysis (MCDA) [34]. This method explicitly evaluates possible solutions in light of conflicting criteria in decision problems. Since there may be complex interactions yielding multiple solutions with conflicting objectives, it is useful to score these solutions according to criteria spawning from sources as diverse as patient preferences, severity of disease for which recommendations are advised, benefit/risk analysis, and so forth. However, for the given example, we will consider the similarity between drugs as the mitigation principle used. The following algorithm 1 is responsible for determining the interactions between the different drugs of the guidelines as well as determining the set of alternatives having as a mitigation criterion the similarity between drugs, calculating this similarity using as input the similarity score provided by the RxNorm API as described in section 3.2. In the first step, the algorithm tries to find



Recomendation interactions

There is a possibility of some drug-drug interactions can impair the patient state. See details below.

The medication Leuprolide interacts with Insulin. The ingredient Insulin is being applied in the task **Monitoring Glucose Control.**.

Severity: N/A

Description: The therapeutic efficacy of Insulin Human can be decreased when used in combination with Leuprolide.

The medication Goserelin interacts with Insulin. The ingredient Insulin is being applied in the task **Monitoring Glucose Control.**.

Severity: N/A

Description: The therapeutic efficacy of Insulin Human can be decreased when used in combination with Goserelin.

The medication Histrelin interacts with Insulin. The ingredient Insulin is being applied in the task **Monitoring Glucose Control.**.

Severity: N/A

Description: The therapeutic efficacy of Insulin Human can be decreased when used in combination with Histrelin.

The medication Triptorelin interacts with Insulin. The ingredient Insulin is being applied in the task **Monitoring Glucose Control.**.

Severity: N/A

Description: The therapeutic efficacy of Insulin Human can be decreased when used in combination with Triptorelin.

Recomendation alternatives

Fig. 8 Recommendation interactions for the given case test in the CompGuide Personal Assistant Web Application.

the conflicts between drug A and B. If a drug conflict exists, the algorithm finds alternative drugs by calling the RxNorm API, according to section 3.2. Later, it calculates the high similarity score provided by RxNorm API [6] for the alternatives of drug A and B. For each alternative with the higher score; it tries to find if a drug conflict exists. If there is a conflict, the algorithm finds the next alternative with the higher score, if there is no conflict, it stores the alternative in the database. After, the system displays the alternative drugs.

Based on the mentioned algorithm and based on the given case example, the reproduced recommendation alternatives are shown in Figure 9.

5 Conclusions and Future Work

Although current CIG solutions offer ways to execute, represent and acquire CIGs, they lack in providing mechanisms to allow a workflow from creation to execution. Moreover, they lack in functionalities such as scheduling and temporal management of clinical protocols, the combination of clinical protocols, user-friendly tools to create and edit of CIGs. CompGuide is able to cover this aspect by providing a unified pipeline for a CIG life cycle.

An essential aspect in the development of CDSSs is the support that these systems can give to health professionals, which are subject to stressful situations, responsible for medical errors, variations in clinical practice, and practice of defensive medicine. This shows that it is necessary to approach health professionals with good clinical practice and evidence-based medicine, by giving some assistance in the decision-making with the help of computer science.

Algorithm 1 Find recommendation drug interactions for each pairwise combination and provide alternative tasks.

Variable: $Drug_A$: a specific drug of a recommendation A
Variable: $Drug_B$: a specific drug of a recommendation B
Variable: $ALTs_A$: the set of alternative recommendations of recommendation A
Variable: $ALTs_B$: the set of alternative recommendations of recommendation B
Variable: ALT_A: the alternative drug A with higher similarity score to drug A
Variable: ALT_B: the alternative drug B with higher similarity score to drug B
Variable: INT_{AB}: a boolean value that describes if drug A and B have a conflict
Variable: Sim_A: the similarity score of Drug A
Variable: Sim_B: the similarity score of Drugs B

findInteraction($Drug_A$**,** $Drug_B$**)**: The function determines if there is an interaction between drug A and drug B.

findAlternativeDrugs($Drug_A$**)**: The function returns a list of alternative drugs to drug A, retrieved from RxNorm.

similarityScore($Drug_A$**)**: The function that returns the similarity score between drug A and the alternative drug, returned by RxNorm API.

storeAlternativeDrug($Drug_A$**)**: The function that allows storing the alternative drug A in the database.

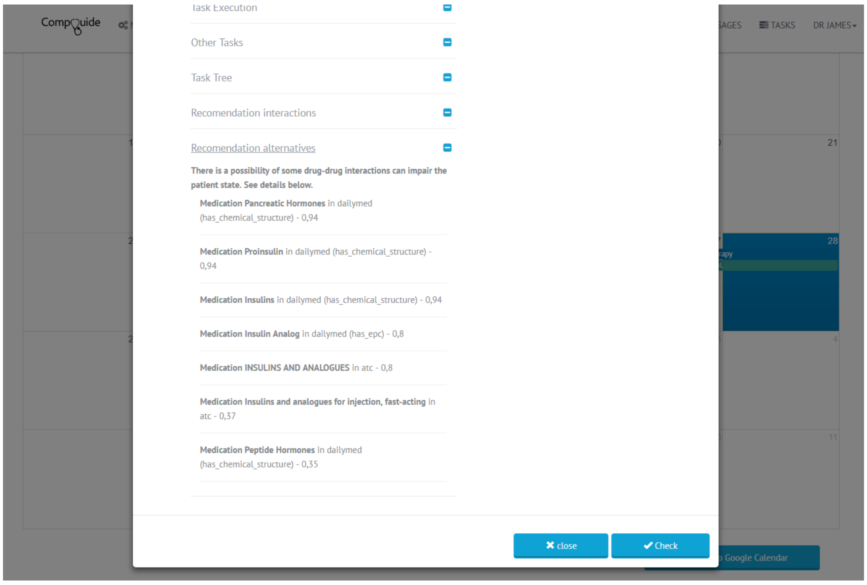
```
$INT_{AB} \leftarrow findInteraction(Drug_A, Drug_B)$   ▷ Find the interaction between drug A and B
if $INT_{AB} == true$ then   ▷ A drug conflict exists
    $ALTs_A \leftarrow findAlternativeDrugs(Drug_A) \cup ALTs_A$   ▷ Find alternative drugs by calling the RxNorm API
    $ALTs_B \leftarrow findAlternativeDrugs(Drug_B) \cup ALTs_B$

end if
$ALT_A \leftarrow getAlternativeHighSimilarityScore(ALTs_A)$   ▷ Get high score alternative drug
$ALT_B \leftarrow getAlternativeHighSimilarityScore(ALTs_B)$

$Sim_A \leftarrow similarityScore(ALT_A)$   ▷ Get similarity score for each alternative drug
$Sim_B \leftarrow similarityScore(ALT_B)$
if $Sim_A$ ¿ $Sim_B$ then   ▷ Determine which alternative will be used by comparing the similarity scores.
                                                  ▷ Alternative drug A was selected
    $INT_{ALT_A}Drug_B \leftarrow findInteraction(ALT_A, Drug_B)$   ▷ Determine if there is a conflict between the drug and the alternative
    while $INT_{ALT_A}Drug_B == true$ —— $ALT_A \neq null$ do   ▷ If there is a conflict, try to find another alternative
        $ALT_A \leftarrow getNextAlternativeHighSimilarityScore(ALTs_A)$
        $INT_{ALT_A}Drug_B \leftarrow findInteraction(ALT_A, Drug_B)$
    end while

    if $ALT_A \neq null$ then
        $storeAlternativeDrug(ALT_A)$   ▷ Store in database the alternative
    end if
else   ▷ Alternative B was selected

    $INT_{ALT_{B}Drug_A} \leftarrow findInteraction(ALT_B, Drug_A)$
    while $INT_{ALT_{B}Drug_A} == true$ —— $ALT_B \neq null$ do
        $ALT_B \leftarrow getNextAlternativeHighSimilarityScore(ALTs_B)$
        $INT_{ALT_{B}Drug_A} \leftarrow findInteraction(ALT_B, Drug_A)$
    end while
    if $ALT_B \neq null$ then
        $storeAlternativeDrug(ALT_B)$
    end if
end if
$displayAlternativeDrugs()$   ▷ Display the alternatives
```



There is a possibility of some drug-drug interactions can impair the patient state. See details below.

Medication Pancreatic Hormones in dailymed (has_chemical_structure) - 0,94

Medication Proinsulin in dailymed (has_chemical_structure) - 0,94

Medication Insulins in dailymed (has_chemical_structure) - 0,94

Medication Insulin Analog in dailymed (has_epc) - 0,8

Medication INSULINS AND ANALOGUES in atc - 0,8

Medication Insulins and analogues for injection, fast-acting in atc - 0,37

Medication Peptide Hormones in dailymed (has_chemical_structure) - 0,35

Fig. 9 Recommendation alternatives for the given case example in the CompGuide Personal Assistant Web Application.

We provide a system that allows to combine the knowledge of several guidelines and to identify drug interactions and conflicts among many recommendations automatically. Moreover, we provide a mechanism that allows a workflow from creation to the execution of CIGs that other systems do not cover. Comparing with the existing systems for CIG execution, they have limitations since they do not focus on combining the knowledge between guidelines. Thus, CompGuide has an additional advantage since it presents a method to identify drug-drug conflicts and provide alternative recommendations automatically. In addition to providing decision support functionalities common to systems for combining CIGs, the CompGuide system offers scheduling and alert features to assist the healthcare professionals in keeping track of their patients. Regarding systems of CIG representation and acquisition, the CompGuide Plugin Editor assists users in creating effective CIGs, by providing information step-by-step on how to fill the data for guideline entries, the capacity to re-utilise knowledge units, and the verification of user inputs.

However, we recognise that it is essential to evaluate the current solution in a clinical environment by testing if the system meets the requirements of the health professionals. So, we intend to do experiments with the physicians to compare the recommendations provided by the system to those the healthcare professional would usually issue, infer about the usefulness of the developed application and obtain an assessment of the fitness of the system to CIG deployment. As future work, we intend to fully implement MCDA in order to allow healthcare professionals to specify different criteria to solve conflicts with medical recommendations, beyond the simple comparison of drug interactions.

This involves not only assessing the benefit-risk of applying the recommendations but also getting patient preferences since some treatment plans can have harmful effects on the patient's health that can alter the quality of life.

acknowledgements

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019. The work of Tiago Oliveira was supported by JSPS KAKENHI Grant Number JP18K18115.

References

1. P.A. de Clercq, J.A. Blom, H.H.M. Korsten, A. Hasman, Artificial intelligence in medicine **31**(1), 1 (2004). DOI 10.1016/j.artmed.2004.02.003. URL http://www.sciencedirect.com/science/article/pii/S0933365704000399
2. K. Kaiser, S. Miksch, Artificial Intelligence in Medicine **46**, 55 (2009). DOI 10.1016/j.artmed.2008.08.009
3. M.N. Hadley, Journal of neurosurgery. Pediatrics **3**(4), 247 (2009). DOI 10.3171/2008.12.PEDS08448. URL http://www.ncbi.nlm.nih.gov/pubmed/19338400
4. P. Novais, R. Costa, D. Carneiro, J. Neves, JAISE **2**(2), 179 (2010)
5. T. Oliveira, A. Silva, J. Neves, P. Novais, Journal of Medical Systems **41**(1), 1 (2017). DOI 10.1007/s10916-016-0655-6. URL http://link.springer.com/10.1007/s10916-016-0655-6
6. S. Liu, W. Ma, R. Moore, V. Ganesan, S. Nelson, IT professional **7**(5), 17 (2005)
7. T. Oliveira, P. Novais, J. Neves, Artificial Intelligence Review pp. 999–1027 (2014)
8. A. Silva, T. Oliveira, F. Gonçalves, J. Neves, K. Satoh, P. Novais, in *World Conference on Information Systems and Technologies* (Springer, 2018), pp. 810–819
9. M. Samwald, K. Fehre, J. de Bruin, K.P. Adlassnig, Journal of Biomedical Informatics **45**(4), 711 (2012). DOI 10.1016/j.jbi.2012.02.001. URL http://dx.doi.org/10.1016/j.jbi.2012.02.001
10. M. Peleg, a.a. Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P. Mork, L. Ohno-Machado, E.H. Shortliffe, R.a. Greenes, in *Proceedings / AMIA ... Annual Symposium. AMIA Symposium* (American Medical Informatics Association, 2000), pp. 645–649. DOI D200640 [pii]
11. M. Balser, C. Duelli, W. Reif, in *Proceedings of the International Conference on Integrated Design and Process Technology* (2002)
12. E. Vier, J. Fox, N. Johns, C. Lyons, A. Rahmanzadeh, P. Wilson, Computer Methods and Programs in Biomedicine **2607**(97) (1997)
13. P. Ram, D. Berg, S. Tu, G. Mansfield, Q. Ye, R.M. Abarbanel, N. Beard, Studies in health technology and informatics **107**(Pt 1), 251 (2004). DOI 10.3233/978-1-60750-949-3-251. URL http://www.ncbi.nlm.nih.gov/pubmed/15360813
14. M.A. Musen, AI matters **1**(4), 4 (2015)
15. N. Beard, J.R. Campbell, S.M. Huff, M. Leon, J.G. Mansfield, E. Mays, J.C. McClay, D.N. Mohr, M.A. Musen, D. O'Brien, et al., in *AMIA* (2002)
16. D.R. Sutton, J. Fox, Journal of the American Medical Informatics Association **10**(5), 433 (2003)
17. R.N. Shiffman, A. Agrawal, A.M. Deshpande, P. Gershkovich, Studies in health technology and informatics (1), 271 (2001)
18. P. Votruba, *Structured knowledge acquisition for asbru* (na, 2003)
19. M.E. Tinetti, S.T. Bogardus Jr, J.V. Agostini, The New England journal of medicine **351**(27), 2870 (2004)
20. C.M. Boyd, J. Darer, C. Boult, L.P. Fried, L. Boult, A.W. Wu, Jama **294**(6), 716 (2005)

21. S. Dumbreck, A. Flynn, M. Nairn, M. Wilson, S. Treweek, S.W. Mercer, P. Alderson, A. Thompson, K. Payne, B. Guthrie, Bmj **350**(mar11 2), h949 (2015). DOI 10.1136/bmj.h949. URL http://www.bmj.com/cgi/doi/10.1136/bmj.h949
22. S. Wilk, M. Michalowski, W. Michalowski, M.M. Hing, K. Farion, in *AMIA Annual Symposium Proceedings*, vol. 2011 (American Medical Informatics Association, 2011), vol. 2011, p. 944
23. S. Wilk, W. Michalowski, M. Michalowski, K. Farion, M.M. Hing, S. Mohapatra, Journal of biomedical informatics **46**(2), 341 (2013)
24. J.A. López-Vallverdú, D. Riaño, A. Collado, in *Process Support and Knowledge Representation in Health Care* (Springer, 2013), pp. 30–41
25. B. Jafarpour, Dalhousie University (2013)
26. B. Jafarpour, S.S.R. Abidi, in *Conference on Artificial Intelligence in Medicine in Europe* (Springer, 2013), pp. 28–32
27. V. Zamborlini, R. Hoekstra, M. Da Silveira, C. Pruski, A. Ten Teije, F. Van Harmelen, Semantic Web **7**(4), 421 (2016). DOI 10.3233/SW-150212
28. F. Gonçalves, T. Oliveira, J. Neves, P. Novais, in *World Conference on Information Systems and Technologies* (Springer, 2017), pp. 257–266
29. T. Oliveira, P. Novais, J. Neves, in *Trends in Practical Applications of Agents and Multiagent Systems* (Springer, 2013), pp. 77–85
30. T. Oliveira, A. Silva, J. Neves, P. Novais, Journal of medical systems **41**(1), 13 (2017)
31. A. Silva, T. Oliveira, J. Neves, K. Satoh, P. Novais, in *Agents and Multi-Agent Systems for Health Care* (Springer, 2017), pp. 76–93
32. J.L. Mohler, R.T. Lee, E.S. Antonarakis, A.J. Armstrong, A.V. D'Amico, B.J. Davis, T. Dorf, J.A. Eastham, R. Ellis, C.A. Enke, T.A. Farrington, National Comprehensive Cancer Network - Prostate Cancer. Tech. rep., National Comprehensive Cancer Network (2018). DOI 10.1016/B978-0-323-35868-2.00080-3. URL http://linkinghub.elsevier.com/retrieve/pii/B9780323358682000803
33. P. Aschner. New idf clinical practice recommendations for managing type 2 diabetes in primary care (2017)
34. P. Thokala, N. Devlin, K. Marsh, R. Baltussen, M. Boysen, Z. Kalo, T. Longrenn, F. Mussen, S. Peacock, J. Watkins, et al., Value in health **19**(1), 1 (2016)
35. N.F. Noy, M. Crubézy, R.W. Fergerson, H. Knublauch, S.W. Tu, J. Vendetti, M.A. Musen, et al., in *AMIA Annu Symp Proc*, vol. 953 (2003), vol. 953, p. 953

3.4 Providing Alternative Measures for Addressing Adverse Drug-drug Interactions

Title	Providing Alternative Measures For Addressing Adverse Drug-drug Interactions
Authors	António Silva, Tiago Oliveira, Ken Satoh, and Paulo Novais
Publication Type	Conference paper
Chapter	New Knowledge in Information Systems and Technologies
Book series	Advances in Intelligent Systems and Computing
Conference	World Conference on Information Systems and Technologies
Publisher	Springer, Cham
Volume	931
Pages	567-576
Year	2019
Online ISBN	978-3-030-16184-2
DOI	https://doi.org/10.1007/978-3-030-16184-2_54
State	Published online
Scimago journal rank (2019)	0.184, Computer Science Miscellaneous (Q3)

Contribution of the doctoral candidate
The doctoral candidate, António José Linhares da Silva, declares to be the main author and the major contributor of the paper *Providing Alternative Measures For Addressing Adverse Drug-drug Interactions*.

Advances in Intelligent Systems and Computing
DOI 10.1007/978-3-030-16184-2_54

Providing Alternative Measures For Addressing Adverse Drug-drug Interactions

António Silva, Tiago Oliveira, Ken Satoh,
Paulo Novais

Accepted: 30 March 2019

Abstract Clinical Practice Guidelines (CPGs) are documents used in daily clinical practice that provide advice on how to best diagnose and treat diseases in the form of a list of clinical recommendations. When simultaneously applying multiple CPGs to patients, this can lead to complex multiple drug regimens (polypharmacy) with the potential for harmful combinations of drugs. The need to address these adverse drug events calls forth for systems capable of not only automatically represent the common potential conflicts or interactions that can happen when merging CPGs but also systems capable of providing conflict-free alternatives. This paper presents a solution that represents CPGs as Computer-Interpretable Guidelines (CIGs) and allows the automatic identification of drug conflicts and the provision of alternative measures to resolve these conflicts.

Keywords Computer-Intepretable Guidelines · Clinical Decision Support · Ontologies · Drug-drug Interactions · Conflict Resolution.

1 Introduction

Drug-drug interactions occur when an effect of one drug alters the effect of another co-administrated drug [1]. Such interactions are common in *multimorbid* patients since they suffer multiple health conditions and need the application

A. Silva
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: asilva@algoritmi.uminho.pt

T. Oliveira · K. Satoh
National Institute of Informatics, Tokyo, Japan
E-mail: {toliveira, ksatoh}@nii.ac.jp

P. Novais
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: {jneves,pjon}@di.uminho.pt

of different disease-specific treatment plans. To help ease the burden of health care professionals, Clinical Practice Guidelines (CPGs) were developed in order to provide patient-specific advice. These documents accumulate and reflect knowledge on how to best diagnose and treat diseases in the form of a list of clinical recommendations. When treating *multimorbid* patients, health care professionals need to retrieve clinical recommendations from multiple chronic disease CPGs. The result is concurrent execution of treatment recommendations from different CPGs, which may cause conflicts. From the combination of these recommendations, several problems can happen, such includes adverse drug events, and increased treatment complexity and cost of treatment [2].

Several projects were developed not only to formally represent CPGs as Computer Interpretable-Guidelines (CIGs). Through the formalisation of CPGs as CIGs it is possible to develop decision support systems that offer a better possibility of affecting clinical behaviour in relation to narrative documents of the corresponding text versions. The representation of CPGs in digital format can have distinct benefits over paper-based CPGs in that they increase flexibility, minimise errors, and generalise the use of CPGs across institutions. However, few works address potential conflicts or interactions that can happen when merging CIGs.

Although some approaches offer ways to represent the conflicts and interactions between concurrent CIGs, they lack in: provision of alternative measures to resolve conflicts in treatment plans, dynamic search for solutions to conflicts outside the existing knowledge base, and provision of methods to rank and select treatment plans. The first contribution of this work is a characterisation of existing approaches to handle the combination of CIGs, especially for *multimorbid* patients. Then, it presents a system that automatically identifies recommendation interactions, conflicts, and provides alternative measures (mainly in the form of alternative drug recommendations) that resolve the identified conflicts.

The paper is organised as follows. Section 2 describes related work regarding systems for combining CIGs. Section 3, presents an architecture for combining CIGs as well as the contributions for the deployment of CIGs in Clinical Decision Support Systems (CDSSs). Section 4 describes the functionalities supporting care with a case example of how the system processes drug-drug interactions and provides alternative measures. Finally, Section 5 presents conclusions about the work developed so far and future work considerations.

2 Related Work

Several formalisms are proposed in the literature that are aimed to represent the conflicts and interactions among different CPGs. They provide various methods to model the conflicts of CPGs into their knowledge base. In this section, we describe different systems that automatically identify the possible interactions between concurrent CPGs for *multimorbid* patients.

Wilk et al. [3] represent CIGs as an activity graph. They use constraint logic programming and combines it with constraint satisfaction problems. By using constraint logic programming, they identify and mitigate possible adverse interactions when applying multiple guidelines on the same patient, namely identifying conflicts associated with potentially contradictory and adverse activities. They provide notification features that inform the healthcare professionals about the possible conflicts during the definition of the treatment plans. This approach provides automatic identification of conflicts, however, it depends on the availability of the information in the knowledge base about the conflicts between both CIGs in the form of constraints and of pre-existing operators to mitigate these conflicts. This requires substantial manual effort for combining CIGs. Thus, in order to provide automatic identification and resolution of conflicts, solutions need to be defined in a *medical background knowledge* as protocol-dependent rules/constraints.

López-Vallverdú et al. [4] used a rule-based methodology in order to identify and reconcile drug conflicts between recommendations of two concurrently executed CIGs. In order to provide a treatment plan without interactions, they utilise a standard terminology called ATC (Anatomical Therapeutic Chemical Classification System for drugs). The outputted treatment plan comprises a set of ATC-codes of medicines, without interactions, which should be prescribed. They manually build knowledge units for the pairwise combination of diseases in their knowledge base. These knowledge units rely on the existence of drug-drug interactions, the presence of a drug which is adverse to a specific disease (drug-disease interaction) and the absence of a necessary medicine for a combination of diseases. Although this approach can only combine CPGs pairwise, it is possible to achieve a final treatment plan for any number of CIGs by combining a pair of CIGs into a general CIG and then combining the latter with a new CIG. This approach requires significant manual effort as each combination has to be hardcoded.

OntoMorph [5] represent guidelines as a collection of ontologies. They use information such as the general domain, the mappings between CPGs and decision rules for simultaneous execution of CPGs that are provided by domain experts. Based on these ontologies, they developed a system capable of merging two concurrent CIGs into a co-morbid personalised guideline. By representing the CIGs as ontologies, it allows retrieving the clinical tasks from the CPG and converts them to computer-interpretable rules in Ontology Web Language (OWL). Using ontologies is one of the possible solutions to CPG representation. It allows the representation of declarative knowledge (medical statements and propositions) and procedural knowledge (workflow structures and actions) as rules. Ontomorph also has a merging representation ontology, which allows capturing merging criteria to achieve the combination of CIGs. By using Semantic Web Rule Language(SWRL) rules, they can identify potential conflicts during the merging process. Since all conditions need to be defined in their model during the merging process, this increases the effort to maintain the system up-to-date and reduces the possibility of sharing knowledge. In their work, some of the identified limitations were not yet entirely addressed, such as potential contradictions between rules, the scalability of the merging model to combine several CIGs, and how the ontology/rules are maintained up-to-date.

The Transition-based Medical Recommendations for Interactions (TMR4I) model is a model that automatically infers the interaction between recommendations [6] by using meta-rules for the identification and reconciliation of three categories of drug conflicts using SPARQL queries (SPARQL is a W3C-standard for semantic queries). Using meta-rules allows defining how a conflict is identified and how similar drugs without interactions and conflicts can be selected as alternatives. The categories of conflicts within CPGs are *repetition interactions*, *contradiction interactions* and *alternative interactions*. The model was extend in [6] with additional interaction types and several measures such as deontic strength, causation belief, and belief strength. This work provides only a representation of conflicts but does not afford reasoning or any form of decision support.

The limitations of above-mentioned approaches include: restrictions in the number of CIGs that can be combined, necessity of all solutions to be available in a knowledge base, and decidability of reasoning mechanisms. In the approaches that require hard-coded solutions, if a conflict that is not accounted for in the knowledge base appears, the reasoning component will not be able to provide a response. Also worth mentioning is that current approaches do not provide support for ranking sets of guideline recommendations that are consistent.

3 An Architecture for CIG Management with CIG Interaction Detection and Resolution

The present work not only aims to provide recommendations to support medical decision-making but also to represent automatically the conflicts and interactions that can happen when merging CIGs. In this work we focus on drug-drug interactions and propose a system capable of automatically identifying recommendation (drug) interactions using existing terminology services, namely the RxNorm API [7]. Once interactions are identified, we provide alternative measures, i.e., alternative drugs to the ones recommended that would not cause any conflict, through a mitigation function. This function calculates the solutions for the identified conflicts using different mitigation principles such as similarity between drugs or user preferences. The architecture is shown in Figure 1. This architecture is a three-level solution that encompasses the following stages for the CIG deployment: representation of CPGs in CIGs, identification of recommendation interactions and provision of recommendation alternatives in case that some recommendations, when applied together, are adverse. The following sections explains the architecture that integrates these three levels.

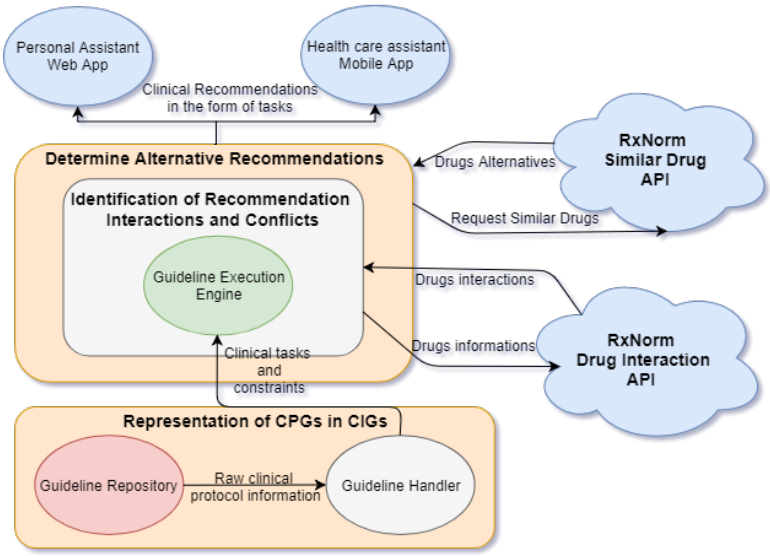

Personal Assistant Web App
Health care assistant Mobile App
Clinical Recommendations in the form of tasks
Determine Alternative Recommendations
Identification of Recommendation Interactions and Conflicts
Guideline Execution Engine
Drugs Alternatives
Request Similar Drugs
RxNorm Similar Drug API
Drugs interactions
Drugs informations
RxNorm Drug Interaction API
Clinical tasks and constraints
Representation of CPGs in CIGs
Guideline Repository
Raw clinical protocol information
Guideline Handler

Fig. 1 Architecture of CompGuide system

3.1 Representation of CPGs in CIGs

The work described herein uses the CompGuide ontology to represent CPGs in the form of a task network. The CompGuide ontology [8] contains different types of clinical tasks such as *Question*, *Action*, *Decision*, *End*, *Plan* and provides different types of clinical constraints expressed in the form of conditions on the patient's state, such as *TriggerConditions*, *PreConditions* and *Outcomes*. Through the utilisation of object properties to connect instances of the sub-classes of the clinical tasks, it is possible to define the relative order between tasks. In the Compguide ontology, it is possible to define sequential tasks, parallel tasks , and alternative tasks. Moreover, it provides a model of temporal representation [9] that aims to represent the temporal constraints placed on clinical tasks. This model represents temporal constructors on the execution of tasks such as *Durations*, *Repetitions*, *Periodicities*, *Waiting Times* and *Repetition Conditions* and temporal constraints about the state of a patient. To acquire and represent CPGs we use the CompGuide plugin which provides information step-by-step on how to fill the data for the guideline entries [10]. This plugin performs the role of managing the creation and editing of CIGs.

The output of guideline encoding is a CIG that will be saved in the *Guideline Repository*. This component is responsible for keeping different CIGs defined according to the CompGuide ontology. The *Guideline Handler* is responsible for managing the access to CIG recommendations in the *Guideline Repository*, providing the clinical tasks and constraints placed on the tasks to the *Guideline Execution Engine*.

3.2 Identification of Recommendation Interactions

The *Guideline Handler* provides clinical task details to the *Guideline Execution Engine* in order to produce recommendations. This component provides information about temporal constraints on the execution of the clinical tasks. Using this information, the *Guideline Execution Engine* produces task enactment times, and by using RxNorm Interaction API, it determines if drug-drug interactions exist. As before-mentioned, the clinical tasks are defined, in the CompGuide ontology, by a set of subclasses. One of these subclasses is *Action*, in which it is possible to define a collection of drugs that require to be prescribed. The *Action* class has several subtypes of actions, namely exams, procedures, medication recommendations and simple recommendations [9]. The Recommendation medication provides a set of drugs that can be prescribed and is used by *Guideline Execution Engine* to determine drug-drug interactions. By using the RxNorm interaction API, it is possible to determine drug-drug interactions, without the need to manually define drug interactions in the knowledge base. Its interaction API [7] uses two data sources, ONCHigh and DrugBank and provide information such as source name, severity and description of the interaction. Thus, the *Guideline Execution Engine* processes all the clinical tasks that are being executed, retrieves all drugs and for each pair of drugs calls the RxNorm Interaction API to obtain the severity and description of the interaction.

3.3 Generating Alternative Recommendations

After processing all drug-drug interactions between concurrently executed clinical recommendations, the alternative measures are evaluated by the system. If an adverse drug event exists, the systems automatically try to find alternative recommendations to resolve the conflict. Through a mitigation function, the system determines which alternative recommendations are advised. This function encompasses a set of steps that include the following:

- **Step 1:** The system tries to find if it is possible to get alternative recommendations, i.e., alternative drugs, in the guidelines. For each specific guideline recommending a drug, the system calculates if an alternative recommendation exists within the guideline. If it is not possible to retrieve the alternative recommendation, the system moves to step 2;
- **Step 2:** Using the RxNorm API, the system tries to find conflict-free alternative drugs. The system provides these alternative drugs by determining the set of alternatives that have the high similarity score concerning the given drugs. The RxNorm API provides the similarity score. If there are no alternative drugs the system moves to step 3;
- **Step 3:** The system evaluates all possible solutions using Multiple-criteria Decision Analysis (MCDA). Since drug-drug interactions are yielding multiple solutions with conflicting objectives, it is useful to score the solutions.

The patient and physician score all possible solutions, so this is a shared patient-clinician evaluation supported by the system.

In step 1, the system tries to resolve the conflicts by analysing within the guideline the different task alternatives. In this particular case, if alternative tasks (recommending alternative drugs) exist in the guideline, the system tries to retrieve them. Then, it gets all the recommended drugs of the alternative tasks and tries to find if drug-drug interactions exist in them, by calling RxNorm Interaction API for each pairwise drugs of the task. In step 2, a ranking of alternative drugs is produced based on the similarity score provided by the RxNorm API. The similarity score between drugs is a score that determines the similarity between drugs. Thus, the system calls the RxNorm API to get alternative drugs for the given conflicted drugs and calculates the highest similarity score for the alternative drugs and for each alternative with the higher score it tries to encounter conflict-free drug. If there is a conflict, the system finds the next alternative with the higher score, if there is no conflict, it stores the alternative in the knowledge base. Table 1 presents the MCDA approach for step 3. This approach uses a value measurement model where for each criterion the patient assigns a score. The objective of this model is constructing and comparing numerical scores (overall value) to identify the degree to which one decision alternative is preferred over another. Each alternative to be scored is a combination of drugs.The system automatically defines the criteria by which decision-makers should orient. Thus, when an adverse drug event occurs, and the system moves to step 3, the criteria established are: severity of disease for which drugs are advised, adverse drug-drug interactions and expected outcomes for the drug application. The criteria are defined on the basis of some types of health care decisions that are implemented in projects such as EMA's Benefit-Risk Methodology Project [11] and shared patient-clinician decision [12]. The total score for each alternative is obtained by multiplying a numerical score for each option on a given criterion by the relative weight for the criterion and later summing these weighted scores. Thus, the total score is provided by the following expression:

$$f(n) = \sum_{n=1}^{n} S^n * WeightC^n \ , \tag{1}$$

where n is the number of solutions to be scored, S^n a score of a specific solution and $WeightC^n$ relative weight for a specific criterion.

The relative weight for the criterion is a value defined by the healthcare professional. This value is requested before starting the evaluation of the solutions. After getting the user scores, the system determines the total score of each solution by using the aforementioned equation. Thus, the total scores of each solution are made available through the Personal Assistant Web App and Healthcare assistant Mobile App, presenting the selected solution. Moreover, after processing the constraints of clinical tasks, determining the interactions between drugs and their alternatives, the clinical recommendations are made available in before-mentioned assistants. In this assistants, it is possible to visualise the clinical recommendations that currently are being applied to the patient, in a calendar and timeline view. Thus, each clinical recommendation can have a set of drugs or alternative solutions that were previously evaluated and scored by decision makers.

Table 1 Assessment of all possible solutions. The symbol C indicates a certain criterion to be evaluated for a given solution α. S means the score of the solution.

Solutions (α)	**Criterion** (C)			**Total Score**
	C^1	...	C^n	
α^1	S^1C^1	...	S^1C^n	f (1) = $\sum_{n=1}^{1} S^1 * WeightC^n$, (2)
...	...	...	...	...
α^n	S^nC^n	...	S^nC^n	f (n) = $\sum_{n=1}^{n} S^n * WeightC^n$, (3)

4 Case Example

This section describes how CompGuide processes the interactions between drugs given a case test example. For this purpose, we used two CIGs based on the NCCN Clinical Practice Guideline for Prostate Cancer [13] and the IDF Clinical Practice Recommendations for managing Type 2 Diabetes [14]. These guidelines were a comprehensive case study since it was possible to test several aspects of the deployment of CIGs . However, in this section, we only address the conflicts between recommendations from many guidelines and provision of alternative recommendations using step 2 described in section 3.2.

For demonstration purposes, we will consider two recommendations from the mentioned guidelines. The first one, named recommendation A belongs to the guideline for managing Type 2 Diabetes: "Apply insulin 0.2 units/kg and titrate once weekly at one unit each time during six months to achieve a target fasting blood glucose between 3.9 and 7.2 mmol/L (70 and 130 mg/dL)". The second recommendation, named recommendation B belongs to the guideline for prostate cancer: "Apply goserelin, leuprolide, histrelin 180 mg/m2 or Triptorelin 100mg/m2 as part of Androgen Deprivation Therapy".

Recommendation A has the action apply insulin, a periodicity value of 1 with a temporal unit of week, a duration value of six, the respective temporal unit of month and medication recommendation insulin. In this case, starting on the 18th of July of 2018 the system will create one event for each week with a duration of one day, during 6 months. The expected conclusion of this task will be on the 18th of January of 2019. As for recommendation B, the action to apply goserelin, leuprolide, histrelin or triptorelin can be identified, with a duration value of 1 and temporal unit of day, starting and finishing on the 18th of July of 2018. The recommendation medications are goserelin, leuprolide, histrelin and Triptorelin. The application tries to provide alternative drugs to address the identified conflicts, by calling RxNorm API and will provide alternative medicines according to step 2, as described in section 3.2. Also, in this step the system calculates a ranking of conflict-free alternative drugs, using the similarity score provided by RxNorm API. The ranking of alternative drugs is calculated by comparing similarity scores and sorting in descending order the medicines according to this similarity values. For the alternative drug (drug provided by RxNorm) with a higher score, the system determines if it is conflict-free over the prescribed drugs. If there is a conflict, the system finds the next alternative with the higher similarity score, if there is no conflict, it stores the alternative in the database and displays the alternative drug as the selected solution.

In the work described herein, we provide a system that automatically identifies conflicts and interactions between drugs for many guidelines. Comparing with the works of Jafarpour et al. (2013) [5], Wilk et al. (2017) [3] and López-Valverdú et al. (2013) [4], where conflicts are defined as constraints in the knowledge base having to be manually specified, CompGuide uses existing terminology services that aggregate different drug sources such as ONCHigh and DrugBank. Thus, through the reuse and integration of existing terminology services such as RxNorm, it is possible to identify conflicts and interactions automatically, without the need to manually define them in the knowledge base. Therefore, using existing terminology services and resorting to external knowledge sources is one of the possible solutions for the limitation mentioned above.Another solution concerns the use of meta-rules such as those used by the TM4I model. Meta-rules can be reused since they can be applied to many CIGs, and conflicts do not need to be manually identified for each guideline, because they can be automatically derived from the guideline representation. However, the bottleneck will be in converting guidelines to computer-interpretable rules. Besides, these systems do not consider aspects such as decision-making. In most cases, there are several alternatives that can lead to conflicting objectives by the decision makers. In other cases, it is necessary to decide which recommendation we want to choose, or which recommendation, in the case at hand, is less adverse. For this specific case, we provide an MCDA approach that allows to evaluate all possible solutions based on conflicting criteria.

5 Conclusions and Future Work

The application of multiple clinical protocols individually can result in complex multiple drug regimens (polypharmacy) with the potential for harmful combinations of drugs. Some of the studied approaches are unable to detect the conflicts for combinations of protocols automatically. Other approaches cannot propose alternative measures that would resolve the conflicts. Other CIG models require all the possible conflicts and their solutions to be available in a knowledge base. Moreover, they cannot lead with cases where decision makers have conflicting solutions or cannot decide on the best treatment alternatives.

As a means to solve these issues, we provide a multiple criteria decision-making approach for not only assessing the benefit-risk of applying the recommendations but also getting patient preferences on best treatment alternatives. This allows to evaluate all possible solutions and to specify different criteria to solve conflicts with medical recommendations, beyond the simple comparison of drug interactions. We also offer a system that allows to combine the knowledge of several guidelines and to identify drug interactions and conflicts among many recommendations automatically. Comparing with some of the studied systems, the CompGuide has an additional advantage since it presents a method to automatically identify drug-drug conflicts among many recommendations, without a necessity of manually define them in the knowledge base. Also, when decision makers have conflicting solutions and cannot decide on the best treatment alternatives, the CompGuide presents an approach that allows to evaluate all possible solutions and to specify different criteria to solve conflicts with medical recommendations. As future work, we intend to make a proper assessment of the fitness of the system for CIG deployment, by performing a study involving physicians interacting with the system in the clinical environment. In this ways, it is possible to analyse if the system meets the requirements of health professionals and if it is user-friendly.

Acknowledgements

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019.

References

1. C. Kennedy, L. Brewer, D. Williams, Medicine (United Kingdom) **44**(7), 422 (2016). DOI 10.1016/j.mpmed.2016.04.015. URL http://dx.doi.org/10.1016/j.mpmed.2016.04.015
2. C.M. Boyd, J. Darer, C. Boult, L.P. Fried, L. Boult, A.W. Wu, Jama **294**(6), 716 (2005)
3. S. Wilk, M. Michalowski, W. Michalowski, D. Rosu, M. Carrier, M. Kezadri-Hamiaz, Journal of biomedical informatics **66**, 52 (2017)
4. J.A. López-Vallverdú, D. Riaño, A. Collado, in *Process Support and Knowledge Representation in Health Care* (Springer, 2013), pp. 30–41
5. B. Jafarpour, S.S.R. Abidi, in *Conference on Artificial Intelligence in Medicine in Europe* (Springer, 2013), pp. 28–32
6. V. Zamborlini, M. Da Silveira, C. Pruski, A. ten Teije, E. Geleijn, M. van der Leeden, M. Stuiver, F. van Harmelen, Artificial intelligence in medicine **81**, 78 (2017)
7. S. Liu, W. Ma, R. Moore, V. Ganesan, S. Nelson, IT professional **7**(5), 17 (2005)
8. T. Oliveira, P. Novais, J. Neves, in *Trends in Practical Applications of Agents and Multiagent Systems* (Springer, 2013), pp. 77–85
9. T. Oliveira, A. Silva, J. Neves, P. Novais, Journal of medical systems **41**(1), 13 (2017)
10. F. Gonçalves, T. Oliveira, J. Neves, P. Novais, in *World Conference on Information Systems and Technologies* (Springer, 2017), pp. 257–266
11. L.D. Phillips, B. Fasolo, N. Zafiropoulos, A. Beyer, Drug discovery today: technologies **8**(1), e3 (2011)
12. J.G. Dolan, E. Boohaker, J. Allison, T.F. Imperiale, Medical Decision Making **33**(1), 59 (2013)

13. J.L. Mohler, R.T. Lee, E.S. Antonarakis, A.J. Armstrong, A.V. D'Amico, B.J. Davis, T. Dorf, J.A. Eastham, R. Ellis, C.A. Enke, T.A. Farrington, National Comprehensive Cancer Network - Prostate Cancer. Tech. rep., National Comprehensive Cancer Network (2018). DOI 10.1016/B978-0-323-35868-2.00080-3. URL http://linkinghub.elsevier.com/retrieve/pii/B9780323358682000803
14. P. Aschner. New idf clinical practice recommendations for managing type 2 diabetes in primary care (2017)

3.5 Mapping a Clinical Case Description to an Argumentation Framework: A Preliminary Assessment

Title	Mapping a Clinical Case Description to an Argumentation Framework: A Preliminary Assessment
Authors	Ana Silva, António Silva, Tiago Oliveira, and Paulo Novais
Publication Type	Conference paper
Conference	International Conference on Intelligent Data Engineering and Automated Learning
Chapter	Intelligent Data Engineering and Automated Learning – IDEAL 2020
Book series	Lecture Notes in Computer Science
Publisher	Springer, Charm
Volume	12489
Pages	338-349
Year	2020
Online ISBN	978-3-030-62362-3
DOI	https://doi.org/10.1007/978-3-030-62362-3_30
State	Published online
Scimago journal rank (2020)	0.249, Computer Science (Miscellaneous) (Q3)

Contribution of the doctoral candidate
The doctoral candidate, António José Linhares da Silva, declares to be one of the main authors and a significant contributor of the paper *Mapping a Clinical Case Description to an Argumentation Framework: A Preliminary Assessment*.

Lecture Notes in Computer Science
DOI 10.1007/978-3-030-62362-3_30

Mapping a Clinical Case Description to an Argumentation Framework: A Preliminary Assessment

Ana Silva · António Silva · Tiago Oliveira · Paulo Novais

Accepted: 27 October 2020

Abstract Medical reasoning in the context of multiple co-existing diseases poses challenges to healthcare professionals by demanding a careful consideration of possible harmful interactions. Computational argumentation, with its conflict resolution capabilities, may assist medical decisions by sorting out these interactions. Unfortunately, most of the argumentation work developed for medical reasoning has not been widely applied to real clinical sources. In this work, we select ASPIC+G and formalise a real clinical case according to the definitions of this argumentation framework. We found limitations in the representation of a patient's evolution and the formalisation of clinical rules which can be inferred from the context of the clinical case.

Keywords Conflict resolution · Computational Argumentation · Medical Reasoning

1 Introduction

In medical reasoning, a healthcare professional establishes a connection between observable phenomena and medical concepts that explain such phenomena [1] [2]. This process involves the integration of clinical information, medical knowledge and contextual factors. The flow of reasoning is guided by medical knowledge which consists of the set of heuristics that use evidence and

Ana Silva · A. Silva
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: {ana.paula.silva, asilva}@algoritmi.uminho.pt

T. Oliveira
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: tiago.jose.martins.oliveira@gmail.com

P. Novais
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal
E-mail: pjon@di.uminho.pt

observations as antecedents and conclude diagnoses and/or next steps. A particularly challenging context for medical reasoning is that of multimorbidity [3], characterised by the co-existence of multiple health conditions in a patient. The difficulties posed by multimorbidity are mainly related to drug-drug and drug-disease interactions. The fist occurs when different drugs prescribed to address different health conditions interact and cause harm to the patient while the latter occurs when a drug prescribed of a health condition causes the aggravation of another existing condition. Hence, the health care professionals must consider not only the observations and clinical evidence in order to recommend treatments to a patient, but also the interactions such recommendations may produce. Another dimension of this process that must be taken into consideration is the preferences of the health care professional and the patient about the recommendations themselves and about the overall goal of the treatment.

Applications of argumentation theory and argumentation frameworks in medical reasoning are not new. We can look as back as early as when Fox and Sutton [4] proposed the PROforma model to find an initial work conveying the usefulness of computational argumentation in a medical setting. However, since then, there have not been works in which the proposed argumentation frameworks are tested and assessed in clinical cases that originate from a different source and with a different structure from the ones used to develop said argumentation frameworks. As such, in this work, we qualitatively assess an argumentation framework, called ASPIC+G [5], proposed for medical reasoning in a context of multimorbidity by applying it to a clinical case extracted from MIMIC III [6], a freely accessible database developed by the MIT Lab for Computational Physiology, comprising the identified health data associated with intensive care unit admissions. Case descriptions include demographics, vital signs, laboratory tests, medications, and medical notes. We focus on the medical notes to identify the clinical information necessary to instantiate the ASPIC+G argumentation framework. The contributions of this work are (i) an analysis of a goal-driven argumentation framework, ASPIC+G, to reason with conflicting medical actions; (ii) an analysis of how information elements in a real clinical case are conveyed and represented in the framework; (iii) the identification of limitations ASPIC+G and an outline of possible to mitigate them. This work doesn't fill in the gap in the literature mentioned above but is intended as a first step in closing that gap.

The paper is organised as follows. In section 2, we provide an overview of argumentation works applied to medical reasoning and highlight the differences between these works and ASPIC+G. Section 3 contains a brief description of the most important components of ASPIC+G, including the representation of clinical information elements. In section 4, we describe the selected clinical case to assess the framework and in section 5 we formalise it in order to build an argumentation theory for medical reasoning. Section 6 conveys the limitations found during representation. Finally, section 7 presents conclusions and future work directions.

2 Related Work

The PROforma modelling language [4] is an executable model language aimed at executing clinical guideline recommendations as tasks. The argumentation component in this work resides in the representation of the core components of decision tasks as arguments for or against a candidate solution. This approach focuses on selecting a task within a single clinical guideline, which, in principle, consists of a set of consistent tasks and does not feature conflicts amongst recommendations. No preferences are considered in argument aggregation in this work.

Another example of reasoning within a single clinical guideline is the approach in [7]. The proposed framework was implemented in the COGENT modelling system and encompasses situations of diagnostic reasoning and patient management. Similarly to PROforma, there are actions that have associated beliefs concerning their effects on the patient state.

A success case in the use of argumentation for medical reasoning is the work in [8]. Therein clinical trials are summarised using argumentation. The framework produces and evaluates arguments that establish the superiority of a treatment over another. These arguments provide conclusions pertaining to a set of outcome indicators and it is possible to establish preferences over these outcomes.

In [9], the authors use argumentation schemes to solve conflicts in recommendations for patient self-management. The argument scheme used is an adaptation of the sufficient condition scheme for practical reasoning which produces an argument in support for each possible treatment. This type of argument leads to the goal to be realised.

Existing approaches generally focus mainly on reasoning within a single set of recommendations [4][7] or do not consider a multimorbidity setting with conflicting recommendations and goals with different priorities [8][9]. Alternatively, the ASPIC+G approach [5] aims to capture these dimensions of clinical reasoning, hence the interest in observing how ASPIC+G handles a case that is different from the one disclosed in that work.

3 ASPIC+G

In this section, we provide an overview of the ASPIC+G argumentation framework as defined in [5] and describe the steps taken to evaluate the framework in light of a real clinical case, extracted from MIMIC III [6].

3.1 Framework Definition

ASPIC+G is an argumentation framework developed to formalise conflict resolution in a medical setting of multimorbidity and compute aggregated consistent sets of clinical recommendations. An argumentation theory in ASPIC+G is a tuple $\langle \mathcal{L}, \mathcal{R}, n, \leqslant \mathcal{R}_d, \mathcal{G}, \leqslant_{\mathcal{G}} \rangle$, where:

- $\mathcal{L}$ is a logical language closed under negation ($\neg$).
- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) rules of the form $\phi_1, \ldots, \phi_n \rightarrow \phi$ and $\phi_1, \ldots, \phi_n \Rightarrow \phi$ respectively, where $n \geq 0$ and $\phi_i, \phi \in \mathcal{L}$;
- n is a partial function s.t. [1] $n : \mathcal{R} \rightarrow \mathcal{L}$;
- $\leqslant_{\mathcal{R}_d}$ is a partial pre-order over defeasible rules $\mathcal{R}_d$, denoting a preference relation, with a strict counterpart $<_{\mathcal{R}_d}$ given by $X <_{\mathcal{R}_d} Y$ iff $X \leqslant_{\mathcal{R}_d} Y$ and $Y \nleqslant_{\mathcal{R}_d} X$;
- $\mathcal{G} \subseteq \mathcal{L}$ is a set of goals that the arguments will try to fulfil s.t. $\forall\ \theta \in \mathcal{G}$, there exists a rule $\phi_1, \ldots, \phi_n \rightarrow \phi$ in $\mathcal{R}_s$ or $\phi_1, \ldots, \phi_n \Rightarrow \phi$ in $\mathcal{R}_d$ s.t. $\phi = \theta$;
- $\leqslant_{\mathcal{G}}$ is a total pre-order on $\mathcal{G}$, denoting *preferences* over goals, with $<_{\mathcal{G}}$ given by $\phi <_{\mathcal{G}} \psi$ iff $\phi \leqslant_{\mathcal{G}} \psi$ and $\psi \nleqslant_{\mathcal{G}} \phi$, and $\simeq_{\mathcal{G}}$ given by $\phi \simeq_{\mathcal{G}} \psi$ iff $\phi \leqslant_{\mathcal{G}} \psi$ and $\psi \leqslant_{\mathcal{G}} \phi$.

Argument construction and argument relations, such as attack and defeat, follow the well-established definitions set by ASPIC+ [10]. One feature provided by ASPIC+G on top of the reasoning mechanisms of ASPIC+ is goal-driven reasoning applied on preferred extensions in order to select a top preferred extension. Let $F = (\mathcal{A}, \mathcal{D}, \mathcal{G}, \leqslant_{\mathcal{G}}, \mathcal{F})$ – where $\mathcal{A}$ is a set of arguments, $\mathcal{D} \subseteq \mathcal{A}$ x $\mathcal{A}$ is a binary relation of defeat, $\mathcal{G}$is a set of goals, $\leqslant_{\mathcal{G}}$ is a preference order over goals, and $\mathcal{F}$ is a binary relation of fulfilment s.t. $\mathcal{F} \subseteq \mathcal{A}$ x $\mathcal{G}$ – be an ASPIC+G argumentation framework and S, a finite set of goals, a preferred extension of F. S is a *top preferred extension of* F iff for every preferred extension S' of F, $\mathrm{Goal}(S') \trianglelefteq_{\mathcal{G}} \mathrm{Goal}(S)$, where S' is another finite set of goals, and defining the goal set ordering, denoted by the operator $\trianglelefteq_{\mathcal{G}}$, as: $S' \trianglelefteq_{\mathcal{G}} S$ iff $S' = \emptyset$ or $\exists g \in (S \setminus S')$ such that $\forall g' \in (S' \setminus S)$, $g' \leqslant_{\mathcal{G}} g$. In the context of a clinical decision, this top preferred extension would the set of treatments selected to be applied to a patient.

3.2 Clinical Information Elements

In the original mapping of clinical information elements to the ASPIC+G three main types of clinical information elements are considered:

- $\mathbb{A}$: the set of all clinical actions which are up for recommendation;
- $\mathbb{E}$: the set containing contraries for all possible effects;
- $\mathbb{S}$: the patient state containing conditions manifested by a patient.

An action $\mathsf{A_x} \in \mathbb{A}$[2] is represented as tuple $\langle \mathsf{t_{x,a}}, \mathsf{O_{x,a}}, \mathsf{P_{x,a}} \rangle$ in which

- $\mathsf{t_x}$ is the treatment conveyed by the action;
- $\mathsf{O_x} = \{(e_1, \mathsf{C_1}, \lambda_1), \ldots, (e_n, \mathsf{C_n}, \lambda_n) : n > 0\}$ is a set of outcomes in which each outcome is a tuple $(e_i, \mathsf{C_i}, \lambda_i), i \in \{1, \ldots, n\}$ brought about by treatment $\mathsf{t_x}$, where: e_i is a description of an effect; $\mathsf{C_i} = \{c_1, \ldots, c_m : m \geq 0\}$

[1] s.t.: such that

[2] Here we omit the second index...

is a set with patient-specific conditions unifiable with the patient state $c_j, j \in \{1, \ldots, m\}$ that enable the occurrence of effect e_i over treatment t_x; λ_i is the impact of an effect e_i, if e_i is a positive effect, then $\lambda_i = \oplus$, otherwise, if it is a negative effect, $\lambda_i = \ominus$.
- $\mathsf{P_x} = \{p_1, \ldots, p_n : n \geq 0\}$ denotes pre-conditions and contains constraints for the application of a treatment $\mathsf{t_x}$.

As an example, let us consider an action that recommends the administration of metformin (*met*) with the intended effect of decreasing glucose levels (*gd*). However, metformin has an undesired side effect which is the acceleration of chronic kidney disease (*ackd*) in patients who have this health condition (*ckd*). Additionally, let us now consider consider an alternative action which recommends the administration of sulfunylurea (*sulf*) to also decrease glucose levels (*gd*). If a patient takes *sulf* he should not take *met* as these drugs have the same effect and their combination could potentially cayuse harm to the patient. The first action would then be represented as $\mathsf{A_1} = \langle met, \{(gd, \emptyset, \oplus), (ackd, \{ckd\}, \ominus)\},$
$\{\neg sulf, \neg met\}\rangle$. Similarly, the second action would be represented as $\mathsf{A_2} = \langle sulf, \{(gd, \emptyset, \oplus), \emptyset\}, \{\neg sulf, \neg met\}\rangle$. We see the pre-conditions used in these actions prevent the simultaneous application of $\mathsf{A_1}$ and $\mathsf{A_2}$.

The next component of multimorbidity management is a set containing the contraries of effects $\mathbb{E} = \{C_1, \ldots, C_n : n \geq 0\}$ where each $C_i, i \in \{1, \ldots, n\}$, is a tuple (e_j, e_k) s.t. $\exists\ \mathsf{A_x} = \langle \mathsf{t_x}, \mathsf{O_x}, \mathsf{P_x}\rangle, \mathsf{A_y} = \langle \mathsf{t_y}, \mathsf{O_y}, \mathsf{P_y}\rangle \in \mathbb{A}$, s.t. $(e_j, \mathsf{C_j}, \lambda_j) \in \mathsf{O_x}$ and $(e_k, \mathsf{C_k}, \lambda_k) \in \mathsf{O_y}$. Expanding the earlier example, let us consider that it would be possible to delay chronic kidney disease by taking another form of medication in addition to one of the previous actions. Considering this new addition to the example effect contraries would take the form $\mathbb{E} = \{(dckd, ackd)\}$.

As for the patient state $\mathbb{S}$, it is defined as a set $\mathbb{S} = \{s_1, \ldots, s_n : n \geq 0\}$ where each $s_i \in S$ is a condition observed or diagnosed in the patient. In the running example, this set would consist of $\mathbb{S} = \{ckd\}$ since there is only one condition the patient is known to have.

3.3 Instantiating the Argumentation Framework

To construct an argumentation theory based on clinical information elements $\mathbb{A}$, $\mathbb{E}$, and $\mathbb{S}$, it is necessary to construct a set of rules $\mathcal{R}$ which will be the backbone of the argumentation theory. The construction of these rules obeys the following specifications:

- $\mathcal{R} = \mathcal{R}_d \cup \mathcal{R}_s$ are respectively defeasible and strict rules in which:
  - $\mathcal{R}_d = \mathcal{R}_1 \cup \mathcal{R}_2$ where $\mathcal{R}_1 = \{\Rightarrow \mathsf{t_{x,a}} \mid \exists \mathsf{A_{x,a}} = \langle \mathsf{t_{x,a}}, \mathsf{O_{x,a}}, \mathsf{P_{x,a}}\rangle \in \mathbb{A}\}$ and $\mathcal{R}_2 = \{\mathsf{t_{x,a}}, c_1, \ldots, c_n \Rightarrow e_z \mid \exists \mathsf{A_{x,a}} = \langle \mathsf{t_{x,a}}, \mathsf{O_{x,a}}, \mathsf{P_{x,a}}\rangle \in \mathbb{A}, (e_z, \{c_1, \ldots, c_n\}, \oplus) \in \mathsf{O_{x,a}}, n \geq 0\}$;
  - $\mathcal{R}_s = \mathcal{R}_3 \cup \mathcal{R}_4 \cup \mathcal{R}_5 \cup \mathcal{R}_6$ where $\mathcal{R}_3 = \{\mathsf{t_{x,a}}, c_1, \ldots, c_n \rightarrow e_z \mid \exists\ \mathsf{A_{x,a}} = \langle \mathsf{t_{x,a}}, \mathsf{O_{x,a}}, \mathsf{P_{x,a}}\rangle \in \mathbb{A}, (e_z, \{c_1, \ldots, c_n\}, \ominus) \in \mathsf{O_{x,a}}, n \geq 0\}$, $\mathcal{R}_4 = \{\mathsf{t_{x,a}} \rightarrow \neg \mathsf{t_{y,b}} \mid \exists\ \mathsf{A_{x,a}} = \langle \mathsf{t_{x,a}}, \mathsf{O_{x,a}}, \mathsf{P_{x,a}}\rangle, \mathsf{A_{y,b}} = \langle \mathsf{t_{y,b}}, \mathsf{O_{y,b}}, \mathsf{P_{y,b}}\rangle \in \mathbb{A}, \neg \mathsf{t_{y,b}} \in$

$\mathsf{P}_{\mathsf{x,a}}\}$, $\mathcal{R}_5 = \{e_j \to \neg e_k \mid (e_j, e_k) \in \mathbb{E}$ or $(e_k, e_j) \in \mathbb{E}$, and $\mathcal{R}_6 = \{\to s \mid s \in \mathbb{S}\}$.

The clinical information elements of our running example would produce the following rules:

- $\mathcal{R}_d = \{\Rightarrow sulf,\ \ \Rightarrow met\} \cup \{sulf \Rightarrow gd,\ met \Rightarrow gd\}$;
- $\mathcal{R}_s = \{met, ckd\} \cup \{sulf \to \neg met\} \cup \{ackd \to \neg dckd\} \cup \{\to ckd\}$;
- $\mathcal{R} = \mathcal{R}_d \cup \mathcal{R}_s$.

There are some important notes about this mapping provided in the original ASPIC+G paper [5]. Disputable facts such as treatments up for selection are represented as defeasible rules with empty antecedents. Additionally, there are two different representations of treatment/effect relationships. A treatment/effect relationship which features a positive effect is represented as a defeasible rule whereas a relationship featuring a negative effect is represented as a strict rule.

It is possible to place preferences on defeasible rules that convey treatments, which allows the users to express preferences for one treatment over another. Adding a scenario in which a patient would manifest a preference for $sulf$ over met would yield the following partial pre-order $\leqslant_{\mathcal{R}_d}$: $(\Rightarrow met) <_{\mathcal{R}_d} (\Rightarrow sulf)$.

In ASPIC+G, it is also possible to specify goals and a total order over these goals which are used in reasoning to select the top preferred extension containing the consistent set of treatments. Goals are set amongst the positive effects of treatments. In the running example, the set of goals would be $\mathcal{G} = \{gd, dckd\}$. Let us add the information that the patient would prioritise *dckd* over *gd*. To convey this preference, the total pre-order over treatment goals is used as follows $\leqslant_{\mathcal{G}}$: $gd <_{\mathcal{G}} dckd$.

4 Case Example

The clinical case selected to map into an ASPIC+G argumentation theory was retrieved from the MIMIC-III database, a publicly-available critical care database [6]. The case was selected from discharge summary reports because they depict complex multimorbidity clinical cases that allow us to test several aspects of ASPIC+G. The case provides a detailed description of the clinical process, which make it possible to easily follow up the inherent medical reasoning.

The clinical case description concerns an 81-year-old female who was admitted to the Medicine service, complaining about a gastrointestinal (GI) haemorrhage (bright red blood per rectum, noticed by blood in her stool and red blood on the paper towel). The patient had an allergy to Penicillins, Vicodin, Cipro and Polysporin. She presented a medical history of atrial fibrillation (A-Fib), diverticulosis and myasthenia gravis. Ex-smoker (quit 25-30 years ago), denied alcohol and drugs. Her father had congestive heart failure, her mother died of myocardial infarction and her siblings had pulmonary fibrosis.

The patient arrived at the hospital after calling her Primary Care Provider, who checked an International Normalized Ratio (INR) which was elevated to 8.0. She went to the hospital where hematocrit (HCT) was 29.0 and INR was 6.1.

On admission, the patient presented sclera anicteric and tachycardia.

During the hospital stay, pericardial effusion was detected and the patient had hypoxia. Lastly, after 5 days of hospitalisation, the patient and her family decided to take comfort measures only (CMO). She was given Morphine and she passed away with her family at her side on the morning of the day after. Below are the medical notes recorded by healthcare professionals.

- **Pericardial effusion:** After a chest x-ray (CXR) it was *noted a left pleural effusion.* Subsequently, in a chest computed tomography (CT) showed *a small to moderate left effusion with a significant pericardial effusion and ascending thoracic aorta aneurysm. (...) A formal echo was performed on the dilated aortic root with probable small sinus of Valsalva aneurysm of the right coronary cusp and severe aortic regurgitation, a moderate circumferential pericardial effusion without frank tamponade.* The cardiology *recommended CT with contrast due to concern for dissection.* The *CT showed dissection of the thoracic ascending aorta with possible rupture into the pericardium.* A *CT surgery was consulted who recommended surgery but the patient declined. She was medically managed with heart rate (HR) and blood pressure control but continued to decline over the next several days with increasing oxygen requirements eventually requiring 100% on a non-rebreather.*
- **Atrial fibrillation with Rapid Ventricular Response (RVR):** *Patient has a history of atrial fibrillation on Coumadin diagnosed* a year ago. *Coumadin was held in the setting of GI bleed. She was given 5 mg IV Metoprolol (x2) and rates slowed to the 120s. She then had a likely vagal episode and became acutely bradycardic with a 6-second pause. The episode resolved spontaneously and the patient reverted back to atrial fibrillation. The patient was transitioned to 25 mg Metoprolol* three times a day (TID) *but remained in A-Fib with RVR. She was placed intermittently on Diltizem drip and converted to normal sinus rhythm (NSR) with rates in the 60s. When the decision was made to make her CMO, these were discontinued.*
- **GI haemorrhage (bright red blood per rectum):** *The patient presents with bright red blood per rectum in the setting of an elevated INR of 8.0 and HCT of 29.0. Her hematocrit dropped to 24.2. She received 2 units* of packed red blood cells (PRBC) *and had an appropriate rise in hematocrit. The INR was corrected with 2 units fresh frozen plasma (FFP) and 5 mg vitamin K x 2. Gastroenterology was consulted who felt this was likely a lower GI bleed, most likely diverticular or arteriovenous malformation (AVM), exacerbated in the setting of elevated INR. Also on the differential diagnosis are haemorrhoids and malignancy. She did have a colonoscopy 2 year ago that did not show any polyps, making malignancy unlikely. The patient's hematocrit stabled and she had no further episodes of bleeding.*

- **Hypoxia:** *Patient with intermittent desaturation. Etiology likely multifactorial secondary to pericardial effusion and poor reserve with underlying myasthenia. The CXR did not show any evidence of acute infection. She was placed on the nasal cannula to maintain oxygen saturation >92%. Please see above, but the patient had increasing oxygen requirements eventually requiring 100% on a non-rebreather. At that time the patient decided to made CMO.*
- **Myasthenia gravis:** *The patient was recently diagnosed with myasthenia after 3 years of progressive symptoms. She had a positive anti-acetylcholine receptor antibody and was started on Pyridostigmine with little improvement. She was recently started on Mestinon. Neurology was consulted regarding diagnosis and treatment and concern for underlying malignancy with paraneoplastic syndrome. They recommended monitoring vital capacity and negative inspiratory force. Neurology weighed in regarding possible surgery and advised that the patient may have a slower recovery coming off of the vent.*

From this clinical case, the information was acquired and treated. Succinctly, the clinical case can be described as:

- GI haemorrhage (*gihem*): to normalize INR (*dinr*) values (which were high), she was given FFP (*ffp*) and 5 mg vitamin K (*vitk*); to increase HCT (*ihct*) was given PRBC (*prbc*).
- Atrial fibrillation (*afib*) with Rapid Ventricular Response (RVR): to prevent a stroke (*ps*), Coumadin (*cou*) was administered; to control heart rate (*chr*), she was given Metoprolol (*met*), but remained in A-Fib with RVR. She was placed intermitantly on Diltizem (*dil*) drip and converted to NSR. These were discontinued when the decision was made to make her CMO (*cmo*).
- Hypoxia (*hyp*): to increase the oxygen intake (*ioi*), patient was ventilated (*vent*).
- Myasthenia gravis (*mg*): She had a positive anti-acetylcholine receptor antibody (*aara*). To relieve symptoms (*rsmg*), the patient took Pyridostigmine (*pyr*) and Mestinon (*mes*). Neurology weighed a possible surgery (*mgsurg*), but the patient may have a slower recovery by leaving ventilation (*vent*).
- Pericardial effusion (*pe*): in a CXR (*cxr*) was noted a left pleural effusion (*lpe*). In a chest CT (*cct*), was remarkable for a small to moderate left effusion with a significant pericardial effusion (*spe*) and ascending thoracic aorta aneurysm (*ataa*). An echo (*echo*) was performed on dilated aortic root with probable small sinus of Valsalva aneurysm (*ssva*) of the right coronary cusp and severe aortic regurgitation (*sar*), a moderate circumferential pericardial effusion (*mcpe*) without frank tamponade. CT with contrast (*ctc*) due to concern for dissection and the CT showed dissection of the thoracic ascending aorta (*dtaa*) with possible rupture into pericardium (*rip*). To treat pericardial effusion (*tpe*), a surgery (*pesurg*) was proposed, but patient refused. The patient and her family made the decision to be made

CMO (*cmo*). She was given Morphine (*mor*) and she passed away with her family at her side.

5 Case Mapping

Considering the multimorbidity clinical case mentioned in section 4, in this section, we perform mapping of its components to ASPIC+G framework. From this case example we have the following actions in $\mathbb{A}$:

$\mathsf{A}_1\langle ffp, \{(dinr, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_2\langle vitk, \{(dinr, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_3\langle prbc, \{(ihct, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_4\langle cou, \{(ps, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_5\langle met, \{(chr, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_6\langle dil, \{(chr, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_7\langle cxr, \emptyset, \emptyset\rangle$;
$\mathsf{A}_8\langle cct, \emptyset, \emptyset\rangle$;
$\mathsf{A}_9\langle echo, \emptyset, \emptyset\rangle$;
$\mathsf{A}_{10}\langle ctc, \emptyset, \emptyset\rangle$;
$\mathsf{A}_{11}\langle pesurg, \{(tpe, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_{12}\langle mor, \{(cmo, \emptyset, \oplus)\}, \{\neg dil, \neg pesurg, \neg mgsurg\}\rangle$;
$\mathsf{A}_{13}\langle vent, \{(ioi, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_{14}\langle aara, \emptyset, \emptyset\rangle$;
$\mathsf{A}_{15}\langle pyr, \{(rsmg, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_{16}\langle mes, \{(rsmg, \emptyset, \oplus)\}, \emptyset\rangle$;
$\mathsf{A}_{17}\langle mgsurg, \{(rsmg, \emptyset, \oplus)\}, \{\neg vent\}\rangle$.

Next step, comprising defining a set of contrary effects $\mathbb{E}$; however, the clinical case example doesn't provide any contrary effect. Thus, the effect contraries are: $\mathbb{E} = \emptyset$.

After, we define the state of the patient as a set of conditions manifested by the patient. We consider $\mathbb{S} = \{lpe, spe, ataa, ssva, sar, paara, mcpe, rip, dtaa\}$.

The set of rules $\mathcal{R}$ that we produce from the clinical case, are as follows:

– $\mathcal{R}_d = \{\Rightarrow ffp, \Rightarrow vitk, \Rightarrow prbc, \Rightarrow cou, \Rightarrow met, \Rightarrow dil, \Rightarrow cxr, \Rightarrow cct, \Rightarrow echo, \Rightarrow ctc, \Rightarrow pesurg, \Rightarrow mor, \Rightarrow vent, \Rightarrow aara, \Rightarrow pyr, \Rightarrow mes, \Rightarrow mgsurg\}$ $\cup$ $\{ffp \Rightarrow dinr, vitk \Rightarrow dinr, prbc \Rightarrow ihct, cou \Rightarrow ps, met \Rightarrow chr, dil \Rightarrow chr, pesurg \Rightarrow tpe, mor \Rightarrow cmo, vent \Rightarrow ioi, pyr \Rightarrow rsmg, mes \Rightarrow rsmg, mgsurg \Rightarrow rsmg\}$;
– $\mathcal{R}_s = \{cmo \rightarrow \neg dil, mgsurg \rightarrow \neg vent, \rightarrow cmo, cmo \rightarrow \neg pesurg\} = \{cmo \rightarrow \neg dil, mgsurg \rightarrow \neg vent, cmo \rightarrow \neg mgsurg, cmo \rightarrow \neg pesurg\}$;
– $\mathcal{R} = \mathcal{R}_d \cup \mathcal{R}_s$;
– $\leqslant \mathcal{R}_d$: $(\Rightarrow pesurg) <_{\mathcal{R}_d} (\Rightarrow mor), (\Rightarrow dil) <_{\mathcal{R}_d} (\Rightarrow mor)$;
– $\mathcal{G} = \{\text{dinr, ihct, ps, chr, tpe, cmo, ioi, rsmg}\}$;
– $\leqslant_{\mathcal{G}}$: dinr $\simeq_{\mathcal{G}}$ ihct $\simeq_{\mathcal{G}}$ ps $\simeq_{\mathcal{G}}$ chr $\simeq_{\mathcal{G}}$ tpe $\simeq_{\mathcal{G}}$ rsmg $<_{\mathcal{G}}$ ioi $<_{\mathcal{G}}$ cmo.

Based on the argument construction rules of ASPIC+G [5] and the goal set described in section 3.3, the arguments $\mathcal{A}$ and goals $\mathcal{G}$ are as follows:

- $\mathcal{A} = \{A_1 :\Rightarrow ffp, A_2 : A_1 \Rightarrow dinr,\ B_1 :\Rightarrow vitk, B_2 : B_1 \Rightarrow dinr,\ C_1 :\Rightarrow prbc, C_2 : C_1 \Rightarrow ihct,\ D_1 :\Rightarrow cou, D_2 : D_1 \Rightarrow ps,\ E_1 :\Rightarrow met, E_2 : E_1 \Rightarrow chr,\ F_1 :\Rightarrow dil, F_2 : F_1 \Rightarrow chr,\ G_1 :\Rightarrow cxr,\ H_1 :\Rightarrow cct,\ I_1 :\Rightarrow echo,\ J_1 :\Rightarrow ctc,\ L_1 :\Rightarrow pesurg, L_2 : L_1 \Rightarrow tpe,\ M_1 :\Rightarrow mor, M_2 : M_1 \Rightarrow cmo,\ M_3 : M_2 \rightarrow \neg dil\ ,\ M_4 : M_2 \rightarrow \neg pesurg,\ M_5 : M_2 \rightarrow \neg mgsurg,\ N_1 :\Rightarrow vent, N_2 : N_1 \Rightarrow ioi,\ O_1 :\Rightarrow aara,\ P_1 :\Rightarrow pyr, P_2 : P_1 \Rightarrow rsmg,\ Q_1 :\Rightarrow mes, Q_2 : Q_1 \Rightarrow rsmg,\ R_1 :\Rightarrow mgsurg, R_2 : R_1 \Rightarrow rsmg,\ R_2' : R_2 \rightarrow \neg vent\}$;
- $\mathcal{G} = \{G_1 : dinr, G_2 : ihct, G_3 : ps, G_4 : chr, G_5 : tpe, G_6 : cmo, G_7 : ioi, G_8 : rsmg\}$

The preferred extensions and their respective goals are as follows:

- $\mathcal{S}_1$ = { $\mathrm{A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2, E_1, E_2, F_1, F_2, G_1, H_1, I_1, J_1, L_1, L_2, N_1, N_2, O_1, P_1, P_2, Q_1, Q_2, R_1, R_2, R_2'}$}, Goal($\mathcal{S}_1$) = {$\mathrm{G_1, G_2, G_3, G_4, G_5, G_7, G_8}$};
- $\mathcal{S}_2$ = { $\mathrm{A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2, E_1, E_2, G_1, H_1, I_1, J_1, M_1, M_2, M_3, M_4, M_5, N_1, N_2, O_1, P_1, P_2, Q_1, Q_2}$}, Goal($\mathcal{S}_2$) = {$\mathrm{G_1, G_2, G_3, G_4, G_6, G_7, G_8}$};

The argumentation theory has two preferred extensions: $\mathcal{S}_1$ - $\mathcal{S}_2$. Admitting the goal ordering established earlier, we calculate the goal set order. There is only one extension $\mathcal{S}_2$ that achieves goal $\mathrm{G_6}$ (the most preferred goal over all the set of goals). Thus, $\mathcal{S}_2$ is the top preferred extension. This means that the decision was made to apply CMO by:

- Administer Morphine;
- Administer FFP and vitamin K to decrease INR;
- Have the patient receive PRBC to increase HCT;
- Administer Coumadin to prevent a stroke;
- Administer Metoprolol to control heart rate;
- Increase the oxygen intake by venting the patient;
- Administer Pyridostigmine and Mestinon to relief symptoms of myasthenia gravis.

6 Discussion

During the mapping of the clinical case, some limitations of the framework were found, particularly concerning temporal events. For example, mapping symptoms over time or even monitoring the evolution of a disease. Moreover, during the construction of rules to ASPIC+G, indirect rules can be derived from a particular clinical case. For instance, it's clearly stated in the mentioned clinical case of section 4 that when the decision was made to CMO, this leads to skipping pericardial effusion surgery. However, there is another rule that we can indirectly set, which is CMO leads to skipping myasthenia gravis surgery. ASPIC+G has a limitation in deriving these indirect rules since it does not provide a formal process of deriving them.

Other information, which is usually relevant in the construction of clinical cases, is the social and family history, which in this framework is also not possible to map. These topics are important to assist health professionals in the diagnosis process and to identify risk factors.

Complementary medicine was mapped as a treatment of an action, with no outcomes and no pre-conditions. This information can be important to validate or discard diseases, helping medical reasoning. It would be important to upgrade ASPIC+G in order to be able to get more information: the reason that led to complementary medicine and the conclusions (if the disease was validated or not, for instance).

To address some of the limitations mentioned above we propose a workflow for extracting the components of the arguments and/or the rules, as depicted in Figure 1. Each clinical case is processed using machine learning techniques (specifically natural language processing techniques) for automatic extraction of treatment, outcomes/effects and the patient status. The main goal will be to use this information to feed an argumentation framework. At the same time, this information can be used in an ontology and its relationships can feed the same framework.

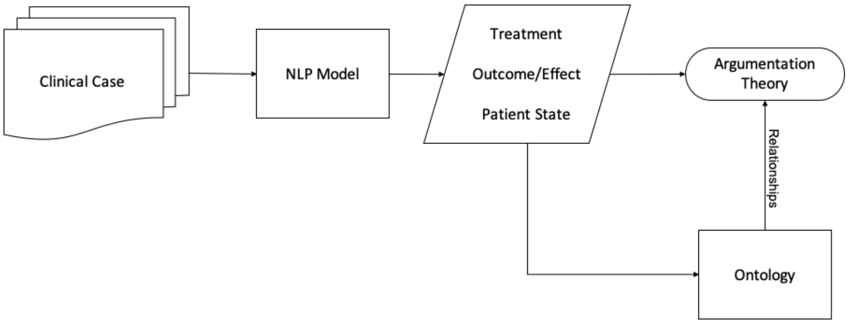


Fig. 1 Workflow to extract argumentation components.

7 Conclusions and Future Work

ASPIC+G has the potential to positively affect clinical behaviour since it provides a framework that properly assesses different conflicting solutions that can arise when treating multimorbidity patients. It provides a formal process to computationally represent clinical actions and their respective components (treatment/effect relations, contrary effects, patient state, and so forth), which provides the necessary information and expressiveness to reasoning in complex scenarios such as the multimorbidity clinical cases. Moreover, its argumentation system offers an alternative to Multiple Criteria Decision Making (MCDM), which allows merging clinical recommendations and use patient-specific goal preferences over treatments to produce the best solution. Furthermore, it has the additional advantage of being more explanatory. However, some points of improvement have been identified that can complete the framework, namely, the inclusion of social and family history, adding information about complementary medicine and map temporal events.

As future work, it is intended to provide an automated mechanism for the extraction of argumentation components (clinical actions, rules and arguments) to include in an argumentation tool. To this end, we will use deep/machine learning techniques [11] such as long short-term memory (LSTM), for natural language processing.

7.0.1 Acknowledgments.

This work has been supported by FCT - Fundação para a Ciência e a Tecnologia within the R&D Units project scope UIDB/00319/2020. The work of Ana Silva is also supported by a Portuguese doctoral grant, SFRH/B D/143512/2019, issued by FCT in Portugal.

References

1. R. Costa, J. Neves, P. Novais, J. Machado, L. Lima, C. Alberto, in *Portuguese Conference on Artificial Intelligence* (Springer, 2007), pp. 323–331
2. D.A. Cook, J. Sherbino, S.J. Durning, Jama **319**(22), 2267 (2018)
3. P. Fraccaro, M.A. Casteleiro, J. Ainsworth, I. Buchan, JMIR Medical Informatics **3**(1), e4 (2015)
4. D.R. Sutton, J. Fox, Journal of the American Medical Informatics Association **10**(5), 433 (2003)
5. T. Oliveira, J. Dauphin, K. Satoh, S. Tsumoto, P. Novais, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **12061 LNAI**, 166 (2020)
6. A.E.W. Johnson, T.J. Pollard, L. Shen, L.w.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, R.G. Mark, Scientific Data **3**(1), 160035 (2016). DOI 10.1038/sdata.2016.35. URL https://doi.org/10.1038/sdata.2016.35
7. M.A. Grando, D. Glasspool, A. Boxwala, Journal of Biomedical Informatics **45**(5), 938 (2012)
8. A. Hunter, M. Williams, Artificial Intelligence in Medicine **56**(3), 173 (2012)
9. N. Kokciyan, I. Sassoon, A.P. Young, M. Chapman, T. Porat, M. Ashworth, V. Curcin, S. Modgil, S. Parsons, E. Sklar, in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence* (2018)
10. S. Modgil, H. Prakken, Argument & Computation **5**, 31 (2014)
11. D. Carneiro, P. Novais, J.M. Pêgo, N. Sousa, J. Neves, in *International Conference on Hybrid Artificial Intelligence Systems* (Springer, 2015), pp. 345–356

Chapter 4

Conclusions

During the doctoral work, the candidate aimed at providing a solution that offers a better possibility of positively affecting clinical behaviour. For this purpose, a CIG-based CDSS solution was developed that integrates features that reflect the concept of *living guidelines*. Thus, it integrates features such as scheduling and temporal management of CPGs, user-friendly tool to create and to edit CIGs, the combination of different disease-specific CPGs and conflict resolution and mitigation between CPGs.

This chapter describes the conclusions of the work developed. We also highlight the objectives achieved and the contributions of this doctoral thesis in light of the objectives outlined. Then, we explain how the research hypothesis was validated by the results achieved. Also, we pointed out the activities performed for the dissemination of results, the work developed so far and future work considerations.

4.1 Contributions of the Doctoral Thesis

In this section, we will present the contributions of this doctoral thesis divided by the different objectives outlined in section 1.6. Also, we present Table 3 showing the objectives and the corresponding sections of chapter 3 where they were achieved. The list of contributions of this doctoral work are as follows:

- **Analysis of the current state of the art for the deployment of CIGs in CDSSs and identification of the main challenges to modelling CPGs as CIGs**

  The first contribution is formalised in section 1.4, which provides a description of the main challenges to formalising CPGs as CIGs and respective integration in CDSSs. This study refers aspects such as complexity regarding the digital representation of CPGs as CIGs, the burden of scheduling the clinical recommendations due to complex temporal constraints and the mapping of the workflow of clinical recommendations, and the conflicts that can happen when applying different disease-specific CPGs. Also, we analyse prominent projects to CIG representation regarding the temporal restrictions in section 3.1 (Related work), and prominent projects for CIG execution in section 3.2 (chapter 2 - Existing Systems for CIG Execution).

Among the different articles, we observed that the decision support features are limited in these CDSS projects, which proved that the existing approaches are still far from the concept of *living guidelines*. We identified limitations in the number of temporal constraint constructors that these models represent, and each studied model of the article in section 3.1, has at least one limitation in one type of temporal constraint. Also, the CDSS approaches lack in information and communication services which transparently support health professionals in their duties and mechanisms for scheduling the clinical recommendations and take control of their enactment times, as referred in section 3.2. Moreover, these solutions lack in providing mechanisms to allow a workflow from creation to execution and processes to identify and mitigate conflicts when merging CPGs, as referenced in section 3.3. The elements included in this contribution accomplish Objective 1 (see Table 3).

- **Identification and characterisation of the requirements of CIG systems that properly fills the identified limitations.**

  The set of methods proposed to address the limitations of CIG systems consists of the using of Ontology Web Language (OWL), for addressing the limitation regarding CIG representation, namely the lack of effectivity of models to represent the temporal constraints of CPGs (section 3.1) and constructors for mapping the workflow from creation to execution of CIGs (3.3). The use of web technologies (section 3.2) allows the development of a more responsive and user-friendly system, which help with the interactivity of the solution. Also, argumentation framework, namely ASPIC+G (section 3.5) and MCDA (section 3.4 and appendix B) were proposed to provide powerful reasoning methods that help health professionals in making well-informed decisions in conflicting clinical situations. These methods provide heuristics that use evidence and observations as antecedents and conclude diagnoses and/or next steps. The idea is to provide automated forms of reasoning about the clinical process and provide proper clinical information to health professionals. This can help to reason in complex scenarios, easing the burden of the clinical practice and help the health professionals to take control of the whole clinical process. Since these decision support features are always available during the clinical process, it is possible to conclude that these promote the pervasiveness of the system. From this contribution, we can conclude that Objective 2 was fully accomplished (see Table 3).

- **Formalisation of a CIG model that provides a comprehensive representation of multiple clinical domains and situations of clinical practice.**

  This contribution is formalised in section 3.1 and then extended in section 3.4 and with more details in appendix B.1 to provide more expressivity by enabling the representation of interactions and their respective mitigation. We use an ontology called CompGuide for CPG representation, which follows a Task Network Model (TNM) and enables the representation of all recommendations, namely *Plans*, *Actions*, *Questions* and *Decisions*. The model allows the expression of administrative information, the definition of clinical tasks' workflow, definition of temporal and clinical constraints. Thus, compa ring with existing models such as Arden Syntax, GLIF3, PROforma or Asbru, CompGuide has more expressivity in the sense that it provides more constructors. Moreover, it has the additional advantage of not requiring proficiency in any programming language. The final ontology has 34 classes, 41 object properties, and 54 data properties. This contribution accomplishes Objective 3 (see Table 3).

- **Definition of architecture for the deployment of CIGs in CDSSs and design an execution engine that handles the workflow of the tasks in the CPGs**

  In section 3.2, the service-oriented architecture of the CompGuide system is presented. As referred in the article presented in the same section, its development as a REST API provides a more flexible way of accessing the CompGuide ontology and functionalities of the guideline execution engine, in different platforms. Thus, it permits the remote guideline execution with data centralisation and facilitates the deployment of CPGs in CDSSs, making them more flexible, portable, and accessible. In a recent development, its API was extended to integrate information about the recommendation interactions and alternatives (section 3.4 chapter 5 and appendix B.2). The definition of the architecture fulfills Objective 4 (see Table 3).

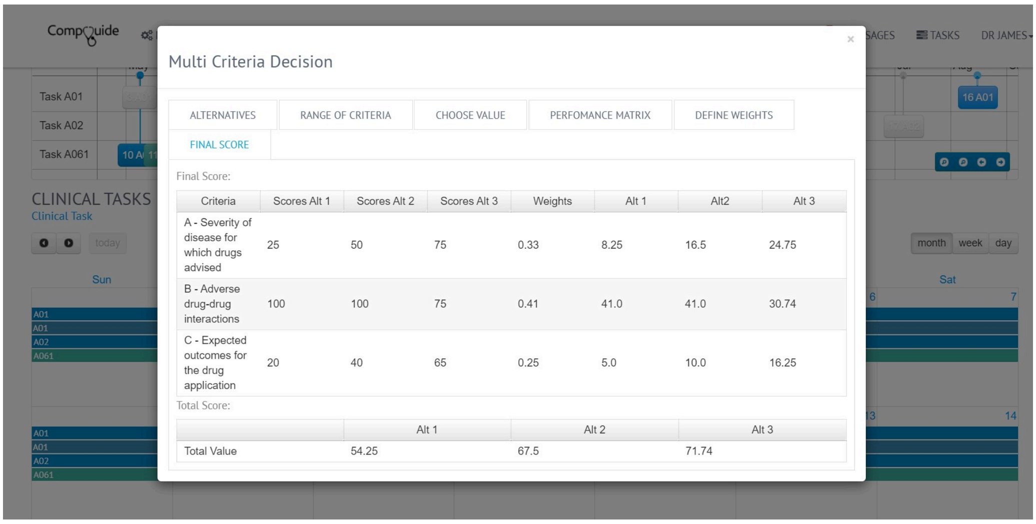
- **Definition of a formalism for the automatic identification and mitigation of the common potential conflicts and interactions that can happen when merging CIGs;**

  The idea herein is to provide a method that automatically identifies conflicts between concurrently executed CPGs and also mitigate these conflicts. For this purpose, an MCDA approach was integrated into CompGuide to assess the benefit-risk of applying concurrently executed recommendations in light of patient preferences on best treatment alternatives, as mentioned in section 3.4 and appendix B. This method integrates linear functions to express patient preferences in scores. Another method was explored, based on argumentation theory, called ASPIG+ that enables the merging of different disease-specific recommendations and use patient preferences to produce the best solutions, as referred in section 3.5. Its advantage when compared to MCDA is that is more explanatory, an important feature in CDSSs. These developments accomplish Objective 5 (see Table 3)

- **Design a platform that permits the visualisation of clinical recommendations and monitors the progress of the clinical process.**

  This contribution is formalised in section 3.2. CompGuide application provides two platforms for the visualisation of clinical recommendations, the Personal Assistant Web Application and the Health Care Assistant Mobile Application. They embed the clinical recommendations of a CIG-based CDSS in activities for health care professionals follow-up. Also, both applications incorporate all the functionalities of data input and reasoning according to the CPG logics, such as scheduling of clinical tasks according to CPG temporal constraints, clinical task ordering according to its relative order in CPG, automatical identification of recommendation interactions and their respective mitigation (using MCDA method). It maps the clinical tasks to an agenda of activities in two forms, namely in a timeline of activities and calendar, which provides different temporal granularities of visualisation. This representation allows to sequence events and reduces the potential for overburdening the health care professional. Moreover, it also contributes to track his activities more efficiently and take control of the whole clinical process while benefiting of automatic reasoning features. In recent works, the tool was expanded to integrate the MCDA model, as can be seen in Figure 14. More details about the new developed interfaces that incorporate the MCDA model are given in appendix A.

  By providing such features, and incorporating CIGs in CDSSs with powerful reasoning features, we aim to assist health professionals in keeping track his tasks and helping them to take control of all clinical process. By doing so, it is possible to consider that Objective 6 was successfully achieved.



Multi Criteria Decision

ALTERNATIVES | RANGE OF CRITERIA | CHOOSE VALUE | PERFOMANCE MATRIX | DEFINE WEIGHTS | FINAL SCORE

Final Score:

Criteria	Scores Alt 1	Scores Alt 2	Scores Alt 3	Weights	Alt 1	Alt2	Alt 3
A - Severity of disease for which drugs advised	25	50	75	0.33	8.25	16.5	24.75
B - Adverse drug-drug interactions	100	100	75	0.41	41.0	41.0	30.74
C - Expected outcomes for the drug application	20	40	65	0.25	5.0	10.0	16.25

Total Score:

	Alt 1	Alt 2	Alt 3
Total Value	54.25	67.5	71.74

Figure 14: Example of output of an MCDA model.

4.2 Validation of the Research Hypothesis

To validate the research hypothesis formulated in section 1.5, we will demonstrate in this section how CompGuide enables the expression of the properties effectiveness, interactivity and pervasiveness, which are related with the construction of a system that calls forth the concept of *living guidelines*.

The effectiveness is conferred by the comprehensive expressivity of the CompGuide ontology, which properly represents important aspects of CPGs. Such includes representation of administrative information, the construction of workflow of clinical tasks, definition of temporal and clinical constraints, and definition of recommendation interactions. These aspects are considered fundamental for a CIG representation and the pre-requisites of an effective CIG model [97]. Given its ontological capacity to manage multiple knowledge patterns and thus represent primitives of different categories and specialities of CPGs, this model shows comprehensive flexibility to adjust to multiple domains.

Regarding the interactivity, this is conferred by the features for supporting care and medical decisions, which in turn help the health professionals to keep track of his clinical tasks and control all aspects of the clinical process. For this propose, the developed solution embeds CIG advice into an agenda of tasks and provides reminders and notifications about important aspects of the execution of the tasks, respecting all temporal and clinical constraints of CPGs. By implementing this communication system with the help of web technologies it was possible to develop a more responsive and user-friendly system. Finally, the idea of expressing such property is to promote the adherence of health care professionals to CPGs and reduce medical error.

On to pervasiveness, this is conferred by the constant availability of the CompGuide system during the clinical process and its capability of providing features for supporting health care professionals and medical decision making. For this purpose, it offers powerful reasoning methods to help health professionals to make well-informed clinical decisions and provide information about all clinical processes to ease their burden and help them to keep track of their tasks and responsibilities.

By integrating these properties in a system, it is possible to enhance CIG-based systems and develop a solution that offers a better possibility of affecting the clinical behaviour and properly address the challenges that clinical practice is currently facing. Thus, it is possible to admit that the research hypothesis was proven.

4.3 Research Activities During Doctoral Programme

As stated in section 1.7, the objective of this doctoral work was reaching the scientific projection and contribute to the enrichment of scientific community knowledge through the dissemination of the results obtained. For this purpose, several publications were written. Along with this task, the co-supervision of a master student, lecturing and the development of other research projects were also conducted. These activities were performed to exchange knowledge and ideas. To demonstrate the reach of the work developed during the doctoral programme, many activities will be described in the next sections.

4.3.1 Other Research Projects

During the doctoral programme, the candidate developed research in other projects, which do not involve the same research field of this doctoral thesis. The first project, called IFactory - Adaptable Standardised Work and Electronic Work Instructions supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within the Project Scope UID/CEC/00319/2013, resulted from a collaboration between the University of Minho and Bosch Braga. The research grant lasted from 15-03-2016 to 31-07-2018 and consisted of the development of a Decision Support System for production planning that includes the following features:

- Design teams of workers on the assembly line with allocation operations (optimisation algorithms);
- Computerisation of the production planning process (PHP Symfony model MVC);
- Decision support system for planning the Standard Work and Work Instructions.

The other project, called Sensible Car - Models for detecting and identifying objects in dynamic environments supported by European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020 POCI-01-0247-FEDER-037902), also resulted from a collaboration between the University of Minho and Bosch Braga. It consisted of the development of a deep learning model, based on convolutional neural networks, for detecting and identifying objects, namely pedestrians, cars, vans, and cyclists, using LiDAR sensing technology. The project provides 3D object detection and its general pipeline detection comprises three stages:

- Data Representation will consume the point clouds to organise this information into a structure that allows the next block to process it more suitably according to the design choices;
- Data Object Detector performs at least two tasks, namely generation of a feature map and detection of the object in this point clouds;
- Multi-task Head network is a multi-task block with the purpose of providing object class prediction and bounding box regression.

4.3.2 Other Publications

In addition to the publications that constitute this doctoral thesis, the doctoral work produced other publications in conference proceedings and international journals. These publications were developed in the fields of Computer Science, Artificial Intelligence and Computer Vision.

4.3.2.1 International Journals

In the context of the Sensible Car research project (project details in section 4.3.1), we submitted an article to a prominent journal. The details are as follows:

- Fernandes, D.*[1], Silva, A.*, Névoa, R.*, Simões, C., Gonzalez, D., Guevara, M., ... & Melo-Pinto, P. (2020). Point-Cloud based 3D Object Detection and Classification Methods for Self-Driving Applications: A Survey and Taxonomy. Information Fusion, https://doi.org/10.1016/j.inffus.2020.11.002.

4.3.2.2 Conference Proceedings

The participation in international conferences resulted in the following publications:

- Silva, A., Oliveira, T., Neves, J., Novais, P. (2017) Transforming Medical Advice into Clinical Activities for Patient Follow-Up. In: Bajo J. et al. (eds) Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems. PAAMS 2017. Communications in Computer and Information Science, vol 722. Springer, Cham, https://doi.org/10.1007/978-3-319-60285-1_14;

- Silva, A., Oliveira, T., Gonçalves, F., Neves, J., Satoh, K., Novais, P. (2018) A Unified System for Clinical Guideline Management and Execution. In: Rocha Á., Adeli H., Reis L., Costanzo S. (eds) Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. Advances in Intelligent Systems and Computing, vol 746. Springer, Cham, https://doi.org/10.1007/978-3-319-77712-2_76;

- Gomes, M., Silva, F., Ferraz, F., Silva, A., Analide, C., Novais, P. (2017) Developing an Ambient Intelligent-Based Decision Support System for Production and Control Planning. In: Madureira A., Abraham A., Gamboa D., Novais P. (eds) Intelligent Systems Design and Applications. ISDA 2016. Advances in Intelligent Systems and Computing, vol 557. Springer, Cham. https://doi.org/10.1007/978-3-319-53480-0_97;

- Silva, A., Oliveira, T., Gonçalves, F., Neves, J., Satoh, K., Novais, P. (2018) A Unified System for Clinical Guideline Management and Execution. In: Rocha Á., Adeli H., Reis L., Costanzo S. (eds) Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. Advances in Intelligent Systems and Computing, vol 746. Springer, Cham. https://doi.org/10.1007/978-3-319-77712-2_76;

- Silva, A., Oliveira, T., Novais, P., and Neves, J., "Representing Temporal Patterns in Computer-Interpretable Clinical Guidelines," inOASIcs - OpenAccess Series in Informatics (C. Schulz andD. Liew, eds.), vol. 49 of OpenAccess Series in Informatics (OASIcs), (Dagstuhl, Germany), p. 69, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. https://doi.org/10.4230/OASIcs.ICCSW.2015.62.

[1] These authors contributed equally to this work

4.3.2.3 Chapter in Books

The work developed in this doctoral work resulted in the publication of a chapter in a book, which is presented below:

- Oliveira, T., Silva, A., Neve,s J., Novais, P. (2016) A Personal Assistant for Health Care Professionals Based on Clinical Protocols. In: Rocha Á., Correia A., Adeli H., Reis L., Mendonça Teixeira M. (eds) New Advances in Information Systems and Technologies. Advances in Intelligent Systems and Computing, vol 444. Springer, Cham. https://doi.org/10.1007/978-3-319-31232-3_80.

4.4 Final Remarks and Perspectives for Future Work

During the doctoral work, it was possible to research many aspects of CDSSs from its representation to the integration of automated reasoning techniques. The results obtained in the case studies implemented show that this area has practical application in the real clinical environment beyond the academic research field. Unfortunately, it was not possible to explore the applicability of the proposed solution in the real clinical setting. Accessing this setting to perform tests can be difficult since healthcare institutions are not always receptive to perform such studies. It implicates changing health care routines, the facility equipment and their layout, and overburdening health professionals since they need to test the system in parallel to performing their duties. Probably in the context of a collaboration between the University of Minho and a health facility, it could be possible to carry out such a task since resources are apriori allocated to perform tests. However, it was possible to consider that the objectives have been satisfactorily achieved, as stated in section 4.1.

As future work, machine learning techniques can be explored to enhance the capability to support decision making. One idea could be the implementation of an automated mechanism for the extraction of argumentation components, using Natural Language Processing methods. Another idea could be to use machine learning algorithms to predict potential drug-drug conflicts using information such as drug phenotypic, therapeutic, chemical, and genomic properties. This can help to anticipate possible drug-drug interactions and thus avoid harmful effects on the patient's health.

As can be seen, these research opportunities demonstrate that there is space to improve CIG-based CDSS systems and approximate CIG advice to the place, time and context of care.

Table 3: Objectives of the doctoral programme and respective document sections where they were achieved.

Objective	Section

Objective 1: Analysis of the current state of art models for the deployment of CIGs in CDSSs and identification of the main challenges to modelling CPGs as CIGs	Section 1.4: Challenges to Modelling CPGs; Section 3.1: Decision Support Provided by a Temporally Oriented Health Care Assistant;
	Section 3.2: A System for the Management of Clinical Tasks Throughout the Clinical Process with Notification Features; Section 3.3: Enhancing Decision Making By Providing A Unified System For CIG Management;

Objective 2: Identification and characterisation of the requirements of CIG systems that properly fills the identified limitations;	Section 3.1: Decision Support Provided by a Temporally Oriented Health Care Assistant;
	Section 3.2: A System for the Management of Clinical Tasks Throughout the Clinical Process with Notification Features;
	Section 3.3: Enhancing Decision Making By Providing A Unified System For CIG Management; Section 3.4: Providing Alternative Measures for Addressing Adverse Drug-drug Interactions;
	Section 3.5: Mapping a Clinical Case Description to an Argumentation Framework: A Preliminary Assessment; Appendix B.

Objective 3: Formalisation of a CIG model that provides a comprehensive representation of multiple clinical domains and situations of clinical practice;	Section 3.1: Decision Support Provided by a Temporally Oriented Health Care Assistant; Section 3.4: Providing Alternative Measures for Addressing Adverse Drug-drug Interactions; Appendix B.1.

Objective 4: Definition of architecture to the deployment of CIGs in CDSSs and design an execution engine that handles the workflow of the tasks in the CPGs;	Section 3.2: A System for the Management of Clinical Tasks Throughout the Clinical Process with Notification Features;
	Section 3.4: Providing Alternative Measures for Addressing Adverse Drug-drug Interactions;
	Appendix B.2.

Objective 5: Definition of a formalism for the automatic identification and mitigation of the common potential conflicts and interactions that can happen when merging CIGs;	Section 3.4: Providing Alternative Measures for Addressing Adverse Drug-drug Interactions; Appendix B; Section 3.5: Mapping a Clinical Case Description to an Argumentation Framework: A Preliminary Assessment.
Objective 6: Design a platform that permits the visualisation of clinical recommendation and monitors the progress of the clinical process.	Appendix A; Section 3.2: A System for the Management of Clinical Tasks Throughout the Clinical Process with Notification Features.

Bibliography

[1] V. Della Mea. “What is e-health (2): The death of telemedicine?” In: *Journal of Medical Internet Research* 3.2 (2001), pp. 6–7. issn: 14388871. doi: 10.2196/jmir.3.2.e22.

[2] G. Eysenbach and T. L. Diepgen. “The role of e-health and consumer health informatics for evidence-based patient choice in the 21st century.” In: *Clinics in Dermatology* 19.1 (2001), pp. 11–17. issn: 0738081X. doi: 10.1016/S0738-081X(00)00202-9.

[3] J.-C. J. Healy. “Implementing e-health in developing countries: Guidance and principles.” In: *ICT Applications and Cybersecurity Division CYB, ...* September (2008), pp. 1–53.

[4] S. Russell and P. Norvig. *Artificial intelligence: a modern approach*. 2nd ed. 2002, p. 1656.

[5] G. F. Luger. *Inteligência Artificial - Estruturas e estratégias para a solução de problemas complexos*. 4th ed. Bookman, 2004, p. 774. isbn: 9798536303962. url: http://books.google.pt/books?id=h2GJLP6jQ4kC&printsec=frontcover#v=onepage&q&f=false.

[6] E. S. Berner and T. J. Lande. *Clinical Decision Support Systems: Theory and Practice*. Springer Science & Business Media, 2007, pp. 64 –98. isbn: 0387383190.

[7] M. A. Musen, Y Shahar, and E. H. Shortliffe. “Clinical decision-support systems.” In: *Biomedical Informatics Computer Applications in Health Care and Biomedicine*. Ed. by E. Shortlife and J. Cimino. Springer, 2006, pp. 698–736.

[8] E. Coiera. *Guide to health informatics*. 3rd ed. CRC Press, 2015, p. 569. isbn: 1444170498.

[9] R. S. Ledley and L. B. Lusted. “Reasoning foundations of medical diagnosis; symbolic logic, probability, and value theory aid our understanding of how physicians reason.” In: *Science (New York, N.Y.)* 130.3366 (1959), pp. 9–21. issn: 0036-8075. doi: 10.1126/science.130.3366.9.

[10] L. Lima, P. Novais, R. Costa, J. B. Cruz, and J. Neves. “Group decision making and Quality-of-Information in e-Health systems.” In: *Logic Journal of the IGPL* 19.2 (2011), pp. 315–332.

[11] L Perreault and J Metzger. “A pragmatic framework for understanding clinical decision support.” In: *Journal of Healthcare Information Management.* 13.2 (1999), pp. 5–21. issn: 1099-811X.

[12] G. Kong, D.-l. Xu, J.-b. Yang, and M. Business. “Clinical Decision Support Systems: A Review on Knowledge Representation and Inference Under Uncertainties.” In: *International Journal of Computational Intelligence Systems* 1.2 (2008), pp. 159–167. issn: 1875-6891. doi: 10.1080/18756891.2008.9727613. url: http://www.tandfonline.com/doi/abs/10.1080/18756891.2008.9727613.

[13] D. F. Sittig, A. Wright, J. A. Osheroff, B. Middleton, J. M. Teich, J. S. Ash, E. Campbell, and D. W. Bates. “Grand challenges in clinical decision support.” In: *Journal of biomedical informatics* 41.2 (2008), pp. 387–392.

[14] R. A. Greenes. *Clinical decision support: the road ahead*. Academic Press, 2011.

[15] J. I. Westbrook, M. T. Baysari, L. Li, R. Burke, K. L. Richardson, and R. O. Day. “The safety of electronic prescribing: manifestations, mechanisms, and rates of system-related errors associated with two commercial systems in hospitals.” In: *Journal of the American Medical Informatics Association* 20.6 (2013), pp. 1159–1167.

[16] K. N. Lohr, M. J. Field, et al. *Clinical practice guidelines: directions for a new program*. Vol. 90. 8. National Academies Press, 1990.

[17] S. Vachhrajani, A. V. Kulkarni, and J. R. W. Kestle. “Clinical practice guidelines.” In: *Journal of Neurosurgery: Pediatrics* 3.4 (2009), pp. 249–256. issn: 1933-0707. doi: 10.3171/2008.12.PEDS08278. url: http://thejns.org/doi/abs/10.3171/2008.12.PEDS08278.

[18] M. D. Cabana, C. S. Rand, N. R. Powe, A. W. Wu, M. H. Wilson, P.-A. C. Abboud, and H. R. Rubin. “Why don’t physicians follow clinical practice guidelines?: A framework for improvement.” In: *Jama* 282.15 (1999), pp. 1458–1465.

[19] M. Lugtenberg, J. S. Burgers, C. F. Besters, D. Han, and G. P. Westert. “Perceived barriers to guideline adherence: a survey among general practitioners.” In: *BMC family practice* 12.1 (2011), p. 98.

[20] M. Samwald, K. Fehre, J. de Bruin, and K. P. Adlassnig. “The Arden Syntax standard for clinical decision support: Experiences and directions.” In: *Journal of Biomedical Informatics* 45.4 (2012), pp. 711–718. issn: 15320464. doi: 10.1016/j.jbi.2012.02.001. url: http://dx.doi.org/10.1016/j.jbi.2012.02.001.

[21] M Peleg, a. a. Boxwala, O Ogunyemi, Q Zeng, S Tu, R Lacson, E Bernstam, N Ash, P Mork, L Ohno-Machado, E. H. Shortliffe, and R. a. Greenes. “GLIF3: the evolution of a guideline representation format.” In: *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*. American Medical Informatics Association, 2000, pp. 645–649.

[22] M. Balser, C. Duelli, and W. Reif. “Formal semantics of Asbru-an overview.” In: *Proceedings of the International Conference on Integrated Design and Process Technology*. 2002.

[23] E Vier, J Fox, N Johns, C Lyons, A Rahmanzadeh, and P Wilson. “PROforma: systems.” In: *Computer Methods and Programs in Biomedicine* 2607.97 (1997).

[24] P. Ram, D. Berg, S. Tu, G. Mansfield, Q. Ye, R. M. Abarbanel, and N. Beard. “Executing clinical practice guidelines using the SAGE execution engine.” In: *Studies in health technology and informatics* 107.Pt 1 (2004), pp. 251–5. issn: 0926-9630. doi: 10.3233/978-1-60750-949-3-251. url: http://www.ncbi.nlm.nih.gov/pubmed/15360813.

[25] T. Oliveira, P. Leão, P. Novais, and J. Neves. “Webifying the Computerized Execution of Clinical Practice Guidelines.” In: *Trends in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection SE - 18* 293 (2014), pp. 149–156. doi: 10.1007/978-3-319-07476-4_18. url: http://dx.doi.org/10.1007/978-3-319-07476-4_18.

[26] F Sonnenberg and C Hagerty. “Computer-interpretable clinical practice guidelines.” In: *Where are we and where are we going* (2006), pp. 145–158.

[27] C. S. Chim, N. T. Cheung, H Fung, and K. C. Wong. “Electronic clinical practice guidelines: current status and future prospects in Hong Kong.” In: *Hong Kong Med J* 9 (2003), pp. 299–301. url: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12904621.

[28] S. Codish and R. N. Shiffman. “A model of ambiguity and vagueness in clinical practice guideline recommendations.” In: *AMIA Annual Symposium Proceedings*. Vol. 2005. American Medical Informatics Association. 2005, p. 146.

[29] S. Dumbreck, A. Flynn, M. Nairn, M. Wilson, S. Treweek, S. W. Mercer, P. Alderson, A. Thompson, K. Payne, and B. Guthrie. “Drug-disease and drug-drug interactions: systematic examination of recommendations in 12 UK national clinical guidelines.” In: *Bmj* 350.mar11 2 (2015), h949–h949. issn: 1756-1833. doi: 10.1136/bmj.h949. url: http://www.bmj.com/cgi/doi/10.1136/bmj.h949.

[30] K. Kaiser and S. Miksch. “Versioning computer-interpretable guidelines: Semi-automatic modeling of ’Living Guidelines’ using an information extraction method.” In: *Artificial Intelligence in Medicine* 46 (2009), pp. 55–66. issn: 09333657. doi: 10.1016/j.artmed.2008.08.009.

[31] J. L. Mohler, R. T. Lee, E. S. Antonarakis, A. J. Armstrong, A. V. D’Amico, B. J. Davis, T. Dorf, J. A. Eastham, R. Ellis, C. A. Enke, and T. A. Farrington. *National Comprehensive Cancer Network - Prostate Cancer*. Tech. rep. National Comprehensive Cancer Network, 2018, 790–799.e2. doi: 10.1016/B978-0-323-35868-2.00080-3. arXiv: arXiv:1011.1669v3. url: http://linkinghub.elsevier.com/retrieve/pii/B9780323358682000803.

[32] P. Aschner. *New IDF clinical practice recommendations for managing type 2 diabetes in primary care*. 2017.

[33] A. Benson, T. Bekaii-Saab, E. Chan, Y.-J. Chen, M. Choti, H. Cooper, and P. Engstrom. *NCCN Clinical Practice Guideline in Oncology Colon Cancer*. Tech. rep. National Comprehensive Cancer Network, 2013. url: http://www.nccn.org/professionals/physician_gls/f_guidelines.asp.

[34] K. Schwaber and M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, p. 158. isbn: 0130676349. url: http://books.google.co.uk/books/about/Agile_Software_Development_with_Scrum.html?id=BpFYAAAAYAAJ&pgis=1.

[35] M. Peleg. “Computer-interpretable clinical guidelines: A methodological review.” In: *Journal of Biomedical Informatics* 46.4 (2013), pp. 744–763. issn: 15320464. doi: 10.1016/j.jbi.2013.06.009. url: http://linkinghub.elsevier.com/retrieve/pii/S1532046413000841.

[36] P. A. de Clercq, J. A. Blom, H. H. M. Korsten, and A. Hasman. “Approaches for creating computer-interpretable guidelines that facilitate decision support.” In: *Artificial intelligence in medicine* 31.1 (2004), pp. 1–27. issn: 0933-3657. doi: 10.1016/j.artmed.2004.02.003. url: http://www.sciencedirect.com/science/article/pii/S0933365704000399.

[37] M. N. Hadley. “Clinical practice guidelines.” In: *Journal of neurosurgery. Pediatrics* 3.4 (2009), 247–8; author reply 248. issn: 1933-0707. doi: 10.3171/2008.12.PEDS08448. url: http://www.ncbi.nlm.nih.gov/pubmed/19338400.

[38] S. H. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw. “Potential benefits, limitations, and harms of clinical guidelines.” In: *Bmj* 318.7182 (1999), pp. 527–530. issn: 0959-8138. doi: 10.1136/bmj.318.7182.527.

[39] R. M. Gardner, T. Pryor, and H. R. Warner. “The HELP hospital information system: update 1998.” In: *International Journal of Medical Informatics* 54.3 (1999), pp. 169–182. issn: 1386-5056. doi: DOI:10.1016/S1386-5056(99)00013-1.

[40] M. A. Musen, S. W. Tu, A. K. Das, and Y. Shahar. “EON: A Component-Based Approach to Automation of Protocol-Directed Therapy.” In: *Emerging Infectious Diseases* 3.6 (1996), pp. 367–388. issn: 10806040. doi: 10.1136/jamia.1996.97084511.

[41] A. Bottrighi, P. Terenziani, S. Montani, M. Torchio, and G. Molino. “Clinical guidelines contextualization in GLARE.” In: *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*. Vol. 2006. American Medical Informatics Association, 2006, p. 860. isbn: 1942-597X (Electronic)\r1559-4076 (Linking). url: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1839666{\&}tool=pmcentrez{\&}rendertype=abstract.

[42] P. D. Clayton, T. A. Pryor, O. B. Wigertz, and G. Hripcsak. “Issues and Structures for Sharing Medical Knowledge among Decision-Making Systems: The 1989 Arden Homestead Retreat.” In: *Annual Symposium on Computer Application in Medical Care*. American Medical Informatics Association, 1989, pp. 116–121. isbn: 0195-4210.

[43] G. Hripcsak, P. Ludemann, T. Pryor, O. B. Wigertz, and P. D. Clayton. “Rationale for the Arden Syntax.” In: *Computers and Biomedical Research* 27.4 (1994), pp. 291–324. issn: 00104809. doi: 10.1006/cbmr.1994.1023. url: http://www.sciencedirect.com/science/article/pii/S0010480984710238.

[44] *HL7 Standards - Master Grid*. 2015. url: http://www.hl7.org/implement/standards/product{_}matrix.cfm?Family=V2 (visited on 08/28/2017).

[45] G. Hripcsak. “Writing Arden Syntax medical logic modules.” In: *Computers in Biology and Medicine* 24.5 (1994), pp. 331–363. issn: 00104825. doi: 10.1016/0010-4825(94)90002-7.

[46] A. Ten Teije, S. Miksch, and P. Lucas. *Computer-based medical guidelines and protocols: a primer and current trends*. Vol. 139. Ios Press, 2008.

[47] L Ohno-Machado, J. H. Gennari, S. N. Murphy, N. L. Jain, S. W. Tu, D. E. Oliver, E Pattison-Gordon, R. a. Greenes, E. H. Shortliffe, and G. O. Barnett. "The guideline interchange format: a model for representing guidelines." In: *Journal of the American Medical Informatics Association : JAMIA* 5.4 (1998), pp. 357–372. issn: 1067-5027. doi: 10.1136/jamia.1998.0050357.

[48] P. E. Stoufflet, L. Ohno-Machado, S. R. A. Deibel, D. Lee, and R. A. Greenes. "GEODE-CM: a state-transition framework for clinical management." In: *AMIA Annual Symposium Proceedings*. Vol. 9. 4. American Medical Informatics Association, 1996, p. 924.

[49] M. Barnes and G. O. Barnett. "An architecture for a distributed guideline server." In: *Proceedings / the ... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*. Vol. 19. IEEE COMPUTER SOCIETY PRESS, 1995, pp. 233–7. isbn: 0195-4210. url: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2579090{\&}tool=pmcentrez{\&}rendertype=abstract.

[50] M Peleg, a. a. Boxwala, E Bernstam, S Tu, R. a. Greenes, and E. H. Shortliffe. "Sharable representation of clinical guidelines in GLIF: relationship to the Arden Syntax." In: *Journal of biomedical informatics* 34.3 (2001), pp. 170–181. issn: 1532-0464. doi: 10.1006/jbin.2001.1016.

[51] S. Miksch, Y. Shahar, and P. Johnson. "Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans." In: *Proceedings of the 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)* (1997). Ed. by E. Motta, F. van Harmelen, C. von Pierret-Golbreich, I. Filby, and N. Wijngaards, pp. 1–25.

[52] Y. Shahar, S. Miksch, and P. Johnson. "The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines." In: *Artificial Intelligence in Medicine* 14.1-2 (1998), pp. 29–51. issn: 09333657. doi: 10.1016/S0933-3657(98)00015-3.

[53] Y. Shahar, O. Young, and E. Shalom. "DEGEL: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines." In: *Artificial Intelligence in ...* Springer, 2003, pp. 122–131. isbn: 978-3-540-20129-8. doi: 10.1007/978-3-540-39907-0_18. url: http://www.springerlink.com/index/YVAE0XVR484WH4GM.pdf.

[54] S. W. Tu, J. R. Campbell, J. Glasgow, M. a. Nyman, R. McClure, J. McClay, C. Parker, K. M. Hrabak, D. Berg, T. Weida, J. G. Mansfield, M. a. Musen, and R. M. Abarbanel. "The SAGE Guideline Model: Achievements and Overview." In: *Journal of the American Medical Informatics Association* 14.5 (2007), pp. 589–598. issn: 10675027. doi: 10.1197/jamia.M2399.

[55] J. Fox, N. Johns, and A. Rahmanzadeh. "Disseminating medical knowledge : the PROforma approach." In: *Lecture Notes in Artificial Intelligence* 14.1 (1997), pp. 157–182. issn: 0933-3657.

[56] A. Vollebregt, A. ten Teije, F. van Harmelen, J. van Der Lei, and M. Mosseveld. "A study of PROforma, a development methodology for clinical procedures." In: *Artificial Intelligence in Medicine* 17.2 (1999), pp. 195–221. issn: 0933-3657. doi: 10.1016/S0933-3657(99)00016-0. url: http.

[57] P. Terenziani, S. Montani, A. Bottrighi, G. Molino, and M. Torchio. "Applying Artificial Intelligence to clinical guidelines: The GLARE approach." In: *Studies in Health Technology and Informatics*. Vol. 139. Springer, 2008, pp. 273–282. isbn: 9781586038731. doi: 10.3233/978-1-58603-873-1-273.

[58] K. P. Adlassnig, C. Combi, A. K. Das, E. T. Keravnou, and G. Pozzi. “Temporal representation and reasoning in medicine: Research directions and challenges.” In: *Artificial Intelligence in Medicine* 38.2 (2006), pp. 101–113. issn: 09333657. doi: 10.1016/j.artmed.2006.10.001.

[59] L. Anselma, P. Terenziani, S. Montani, and A. Bottrighi. “Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines.” In: *Artificial Intelligence in Medicine* 38.2 (2006), pp. 171–195. issn: 09333657. doi: 10.1016/j.artmed.2006.03.007.

[60] A. A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q. T. Zeng, D. Wang, V. L. Patel, R. A. Greenes, and E. H. Shortliffe. “GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines.” In: *Journal of biomedical informatics* 37.3 (2004), pp. 147–61. issn: 1532-0464. doi: 10.1016/j.jbi.2004.04.002. url: http://www.sciencedirect.com/science/article/pii/S1532046404000334.

[61] Y. Shahar, S. Miksch, and P. Johnson. “A task-specific ontology for design and execution of time-oriented skeletal plans.” In: *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, KAW.* Vol. 96. 1996, pp. 1–20. url: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/shahar/.

[62] D. Wang, M. Peleg, S. W. Tu, E. H. Shortliffe, and R. a. Greenes. “Representation of clinical practice guidelines for computer-based implementations.” In: *Studies in Health Technology and Informatics* 84 (2001), pp. 285–289. issn: 09269630. doi: 10.3233/978-1-60750-928-8-285.

[63] P. Terenziani, C. Carlini, S. Montani, P. Orientale, A. Avogadro, and C. Borsalino. “Towards a Comprehensive Treatment of Temporal Constraints in Clinical Guidelines.” In: *Temporal Representation and Reasoning, 2002. TIME 2002. Proceedings. Ninth International Symposium on.* 2002, pp. 20–27.

[64] R. Dechter, I. Meiri, and J. Pearl. “Temporal constraint networks.” In: *Artificial Intelligence* 49.1–3 (1991), pp. 61–95. issn: 0004-3702. doi: 10.1016/0004-3702(91)90006-6. url: http://www.sciencedirect.com/science/article/pii/000437029190006 6{\%}5Cnhttp://www.sciencedirect.com/science/article/pii/0004370291900066/pdf?md5=1d445ab9a8a5dfb9ff1a872ffcbb7ee5{\&}pid=1-s2.0-0004370291900066-main.pdf.

[65] E. H. Sherman, G Hripcsak, J Starren, R. a. Jenders, and P Clayton. “Using intermediate states to improve the ability of the Arden Syntax to implement care plans and reuse knowledge.” In: *Proceedings of the Annual Symposium on Computer Application in Medical Care.* American Medical Informatics Association, 1995, pp. 238–242. isbn: 0195-4210 (Print)\r0195-4210 (Linking).

[66] M. Cristani and R. Cuel. “A survey on ontology creation methodologies.” In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 1.2 (2005), pp. 49–69.

[67] M. A. Musen. “The protégé project: a look back and a look forward.” In: *AI matters* 1.4 (2015), pp. 4–12.

[68] N. F. Noy, M. Crubézy, R. W. Fergerson, H. Knublauch, S. W. Tu, J. Vendetti, M. A. Musen, et al. “Protege-2000: an open-source ontology-development and knowledge-acquisition environment.” In: *AMIA Annu Symp Proc.* Vol. 953. 2003, p. 953.

[69] N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen. “Creating semantic web contents with protege-2000.” In: *IEEE intelligent systems* 16.2 (2001), pp. 60–71.

[70] T.-Y. Leong, K. Kaiser, and S. Miksch. “Free and open source enabling technologies for patient-centric, guideline-based clinical decision support: a survey.” In: *Yearbook of medical informatics* (2007), p. 74.

[71] D. L. Rubin, N. F. Noy, and M. A. Musen. “Protege: a tool for managing and using terminology in radiology applications.” In: *Journal of digital imaging* 20.1 (2007), pp. 34–46.

[72] N. Beard, J. R. Campbell, S. M. Huff, M. Leon, J. G. Mansfield, E. Mays, J. C. McClay, D. N. Mohr, M. A. Musen, D. O’Brien, et al. “Standards-based Sharable Active Guideline Environment (SAGE): A Project to Develop a Universal Framework for Encoding and Disseminating Electronic Clinical Practice Guidelines.” In: *AMIA*. 2002. url: `http://dx.doi.org/10.1093/jigpal/jzq029`.

[73] J Kim, B. Shim, S. Kim, J. Lee, I. Cho, and Y. Kim. “Translation Protégé knowledge for executing clinical guidelines.” In: *Proceedings of Conference Protege*. 2009.

[74] D. R. Sutton and J. Fox. “The syntax and semantics of the PROforma guideline modeling language.” In: *Journal of the American Medical Informatics Association* 10.5 (2003), pp. 433–443.

[75] B. Martínez-Salvador and M. Marcos. “Supporting the Refinement of Clinical Process Models to Computer-Interpretable Guideline Models.” In: *Business & Information Systems Engineering* 58.5 (2016), pp. 355–366.

[76] R. Steele and F. J. T. P. Primer. *introduction to PROforma language and software with worked examples*. Tech. rep. Technical report. London, UK: Advanced Computation Laboratory, Cancer Research, 2002.

[77] E. Lozano, M. Marcos, B. Martínez-Salvador, A. Alonso, and J. R. Alonso. “Experiences in the Development of Electronic Care Plans for the Management of Comorbidities.” In: *KR4HC*. Springer. 2009, pp. 113–123.

[78] O. Ayelet. *The Tallis Composer User Interface*. 2005. url: `http://archive.cossac.org/tallis/pdf/DecisionsCandidatesAndArguments.pdf` (visited on 09/11/2017).

[79] R. N. Shiffman, A. Agrawal, A. M. Deshpande, and P. Gershkovich. “An approach to guideline implementation with GEM.” In: *Studies in health technology and informatics* 1 (2001), pp. 271–275.

[80] G. Michel and R. Shiffman. *GEM Cutter 2.5 User Guide*. 2000. url: `http://gem.med.yale.edu/GEM_CutterII/GEMCutter2.5_UserGuide.pdf` (visited on 09/11/2017).

[81] B. T. Karras, S. Nath, and R. N. Shiffman. “A preliminary evaluation of guideline content mark-up using GEM–an XML guideline elements model.” In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association. 2000, p. 413.

[82] K.-A. Polvani, A. Agrawal, B. Karras, A. Deshpande, and R. Shiffman. “GEM cutter manual.” In: *Yale Center for Medical Informatics* (2000), pp. 122–131.

[83] P. Votruba. *Structured knowledge acquisition for asbru*. na, 2003.

[84] B. Huber. *Asbruview 2.0 User Guide*. 2005. url: `http://www.ifs.tuwien.ac.at/~votruba/AsbruView/Asbruview_User_Guide.pdf` (visited on 09/11/2017).

[85] R Bott. *Summary of the Guideline Workbenches Evaluation*. 2014. url: `http://sage.wherever.org/references/docs/WorkbenchEvaluationSummary.pdf` (visited on 09/11/2017).

[86] K. Marsh, M. Goetghebeur, P. Thokala, and R. Baltussen. *Multi-Criteria Decision Analysis to Support Healthcare Decisions*. Springer, 2017.

[87] P. Thokala, N. Devlin, K. Marsh, R. Baltussen, M. Boysen, Z. Kalo, T. Longrenn, F. Mussen, S. Peacock, J. Watkins, et al. “Multiple criteria decision analysis for health care decision making—an introduction: report 1 of the ISPOR MCDA Emerging Good Practices Task Force.” In: *Value in health* 19.1 (2016), pp. 1–13.

[88] K. Marsh, T. Lanitis, D. Neasham, P. Orfanos, and J. Caro. “Assessing the value of healthcare interventions using multi-criteria decision analysis: a review of the literature.” In: *Pharmacoeconomics* 32.4 (2014), pp. 345–365.

[89] L. D. Phillips, B. Fasolo, N. Zafiropoulos, and A. Beyer. “Is quantitative benefit–risk modelling of drugs desirable or possible?” In: *Drug discovery today: technologies* 8.1 (2011), e3–e10.

[90] X. Kurz. “Advancing regulatory science, advancing regulatory practice.” In: *pharmacoepidemiology and drug safety* 26.6 (2017), pp. 722–726.

[91] J. G. Dolan, E. Boohaker, J. Allison, and T. F. Imperiale. “Patients’ preferences and priorities regarding colorectal cancer screening.” In: *Medical Decision Making* 33.1 (2013), pp. 59–70.

[92] G. Elwyn, D. Frosch, R. Thomson, N. Joseph-Williams, A. Lloyd, P. Kinnersley, E. Cording, D. Tomson, C. Dodd, S. Rollnick, et al. “Shared decision making: a model for clinical practice.” In: *Journal of general internal medicine* 27.10 (2012), pp. 1361–1367.

[93] J. N. Davide Carneiro Paulo Novais. *Conflict Resolution and Its Context: From the Analysis of Behavioural Patterns to Efficient Decision-making*. Springer-Verlag, 2014. isbn: 978-3-319-06238-9. url: http://dx.doi.org/10.1007/978-3-319-06239-6.

[94] A. Guitouni and J.-M. Martel. “Tentative guidelines to help choosing an appropriate MCDA method.” In: *European Journal of Operational Research* 109.2 (1998), pp. 501–521.

[95] H. Saarikoski, D. N. Barton, J. Mustajoki, H. Keune, E. Gomez-Baggethun, and J. Langemeyer. “Multi-criteria decision analysis (MCDA) in ecosystem service valuation.” In: *OpenNESS ecosystem services reference book/Potschin, M.[edit.]; et al.* 2016, pp. 1–6.

[96] Institute of Medicine. *Ethical and Scientific Issues in Studying the Safety of Approved Drugs Committee on Ethical and Scientific Issues in Studying the Safety of Approved Drugs ; Board on Population Health and Public Health Practice ; Institute of Medicine*. 2012. isbn: 9780309218139.

[97] M. Peleg. “Computer-interpretable clinical guidelines: a methodological review.” In: *Journal of biomedical informatics* 46.4 (2013), pp. 744–763.

[98] F. Gonçalves, T. Oliveira, J. Neves, and P. Novais. “CompGuide: Acquisition and editing of computer-interpretable guidelines.” In: *World Conference on Information Systems and Technologies*. Springer. 2017, pp. 257–266.

[99] A. Costa, P. Novais, and R. Simoes. “A caregiver support platform within the scope of an ambient assisted living ecosystem.” In: *Sensors* 14.3 (2014), pp. 5654–5676. url: http://dx.doi.org/10.3390/s140305654.

[100] C. Analide, P. Novais, J. Machado, and J. Neves. "Quality of knowledge in virtual entities." In: *Encyclopedia of Communities of Practice in Information and Knowledge Management*. IGI Global, 2006, pp. 436–442. url: http://dx.doi.org/10.4018/978-1-59140-556-6.ch073.

[101] T. Oliveira, P. Novais, and J. Neves. "Representation of clinical practice guideline components in OWL." In: *Trends in Practical Applications of Agents and Multiagent Systems*. Springer, 2013, pp. 77–85.

[102] S. Liu, W. Ma, R. Moore, V. Ganesan, and S. Nelson. "RxNorm: prescription for electronic drug information exchange." In: *IT professional* 7.5 (2005), pp. 17–23.

[103] T. Oliveira, A. Silva, J. Neves, and P. Novais. "Decision support provided by a temporally oriented health care assistant." In: *Journal of medical systems* 41.1 (2017), p. 13.

[104] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge university press, 1993.

Appendix A

Interfaces Developed to Integrate the MCDA Model

This appendix presents examples of web interfaces, developed to integrate the MCDA model. Figure 15 represents the timeline of the various tasks. Admitting the two actions A02 and A061, depicted in Figure 15, which consist of applying insulin to treat Diabetes Type 2 and applying histrelin to treat prostate cancer, respectively. The first action was extracted from the IDF Clinical Practice Recommendations for managing Type 2 Diabetes [32], and the second from the NCCN Clinical Practice Guideline for Prostate Cancer [31]. These two CIGs were represented in CompGuide ontology using the *CompGuide plugin* [98]. When applying these two recommendations, there is a drug conflict; namely, the drug histrelin harms insulin's therapeutic efficacy. This interaction was obtained using the *RXNorm Interaction API* as described in section 3.4. Table 4 gives a brief description of each recommendation.

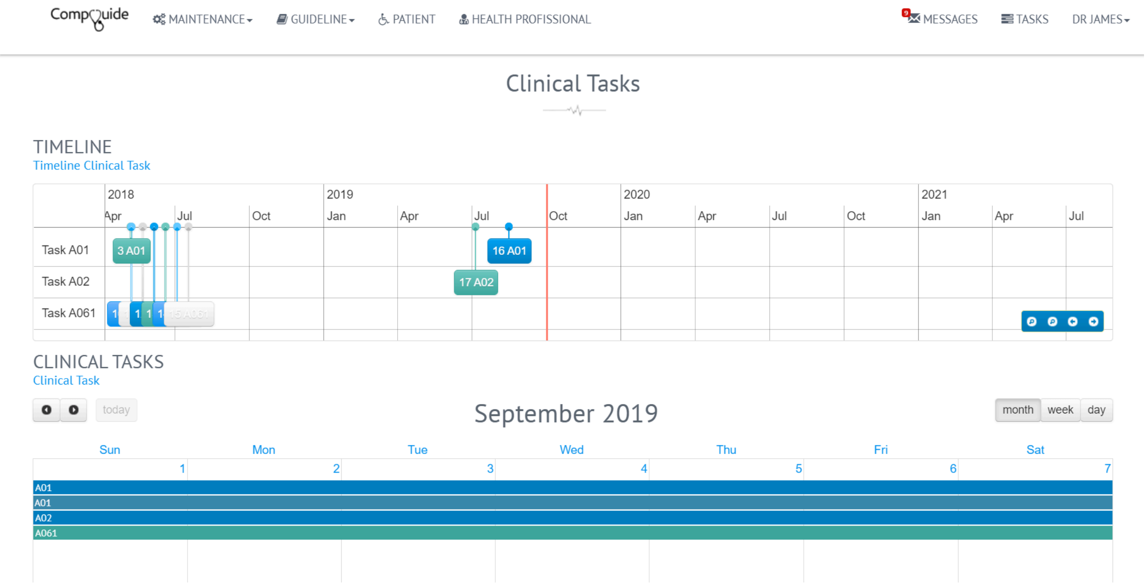
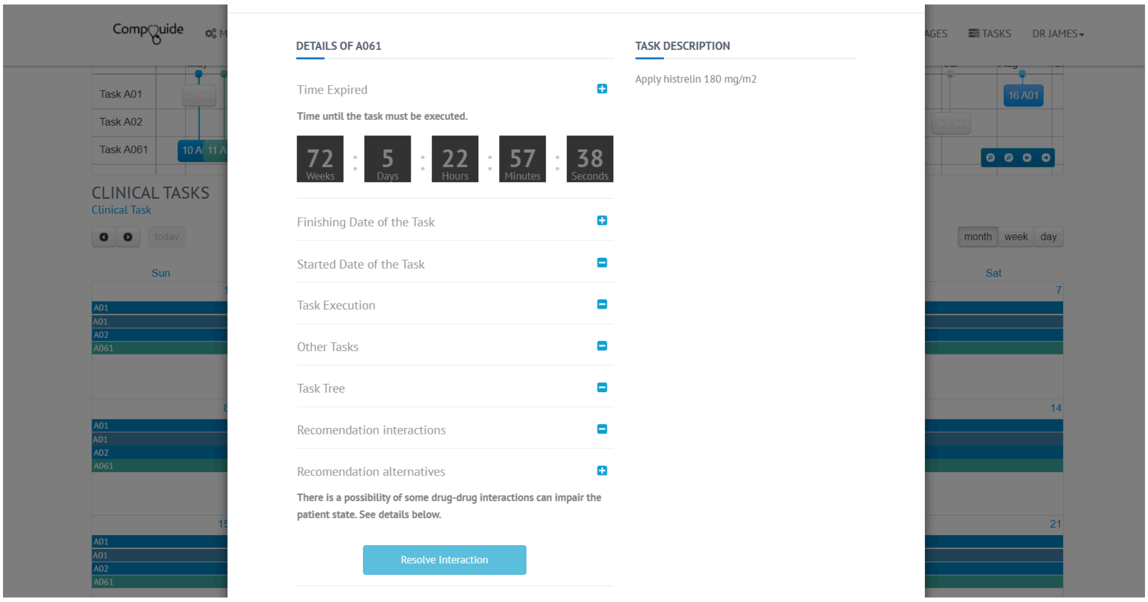


Figure 15: Task Timeline and Scheduling of *Actions* A02 and A061.

Figure 16 shows the web page with the details corresponding to task A61.

Table 4: Description of Recommendations A02 and A061.

Recommendation	Description
A02	Apply insulin 0.2 units/kg and titrate once weekly at one unit each time during six months to achieve a target fasting blood glucose between 3.9 and 7.2 mmol/L (70 and 130 mg/dL).
A061	Apply histrelin 180 mg/m2 as part of Androgen Deprivation Therapy.



DETAILS OF A061

Time Expired ⊞

Time until the task must be executed.

72 Weeks : 5 Days : 22 Hours : 57 Minutes : 38 Seconds

Finishing Date of the Task ⊞

Started Date of the Task ⊟

Task Execution ⊟

Other Tasks ⊟

Task Tree ⊟

Recomendation interactions ⊟

Recomendation alternatives ⊞

There is a possibility of some drug-drug interactions can impair the patient state. See details below.

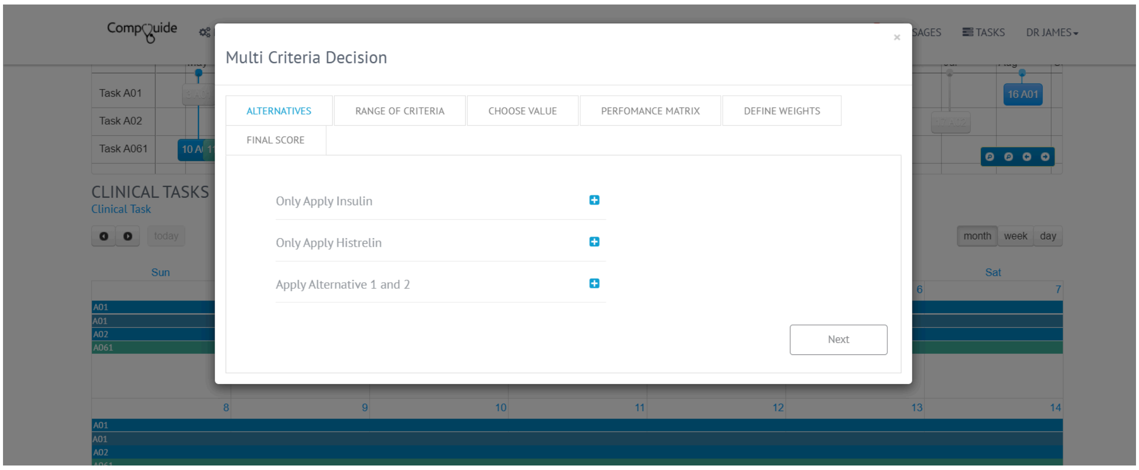
Resolve Interaction

TASK DESCRIPTION

Apply histrelin 180 mg/m2

Figure 16: Task A61 Details.

Figure 17 corresponds to the first tab of the MCDA model web page, which depicts the different alternatives in the given clinical case.



Multi Criteria Decision

ALTERNATIVES | RANGE OF CRITERIA | CHOOSE VALUE | PERFOMANCE MATRIX | DEFINE WEIGHTS | FINAL SCORE

Only Apply Insulin ⊞

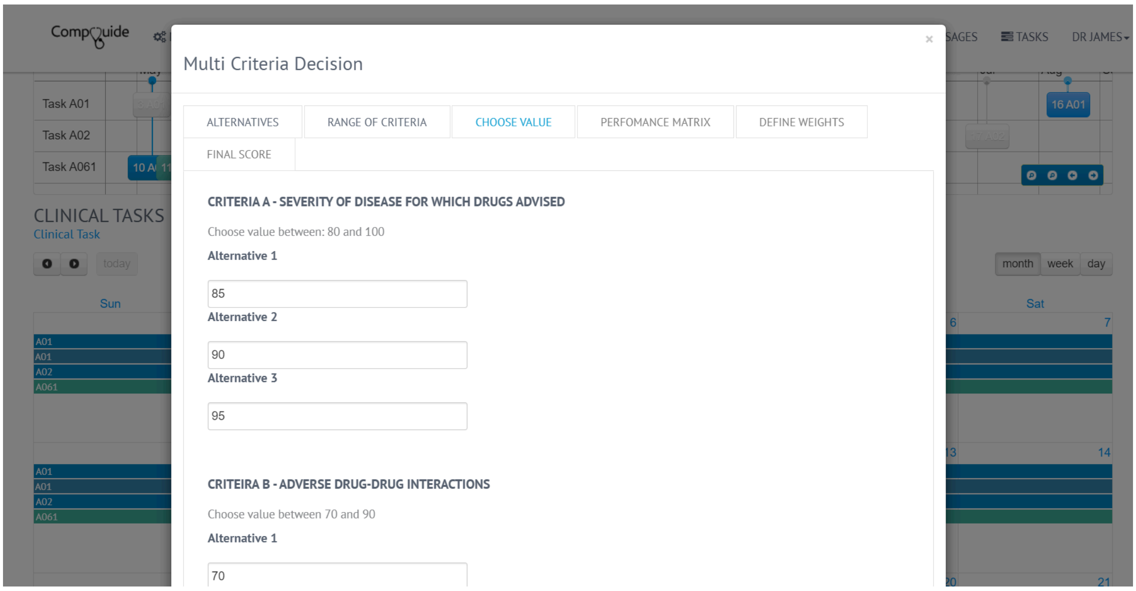
Only Apply Histrelin ⊞

Apply Alternative 1 and 2 ⊞

Next

Figure 17: MCDA model alternatives available for the given clinical case conflict.

After defining the ranges for each criterion, the patient and the healthcare professional must define the importance of each alternative for each criterion. Figure 18 represents the choice of this importance.



CRITERIA A - SEVERITY OF DISEASE FOR WHICH DRUGS ADVISED

Choose value between: 80 and 100

Alternative 1: 85

Alternative 2: 90

Alternative 3: 95

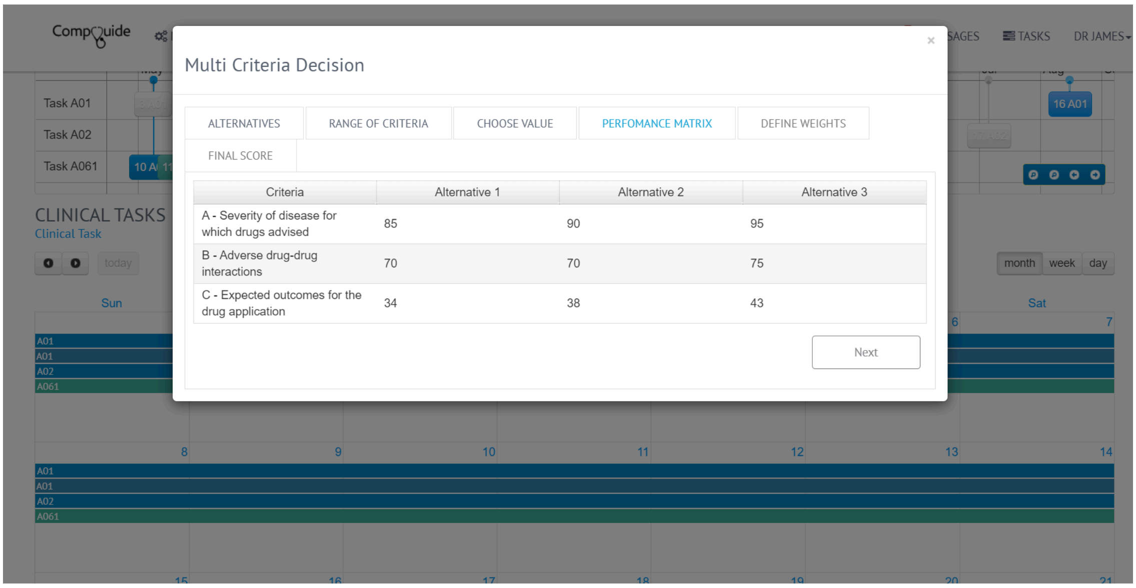
CRITEIRA B - ADVERSE DRUG-DRUG INTERACTIONS

Choose value between 70 and 90

Alternative 1: 70

Figure 18: Importance of alternatives in each criterion.

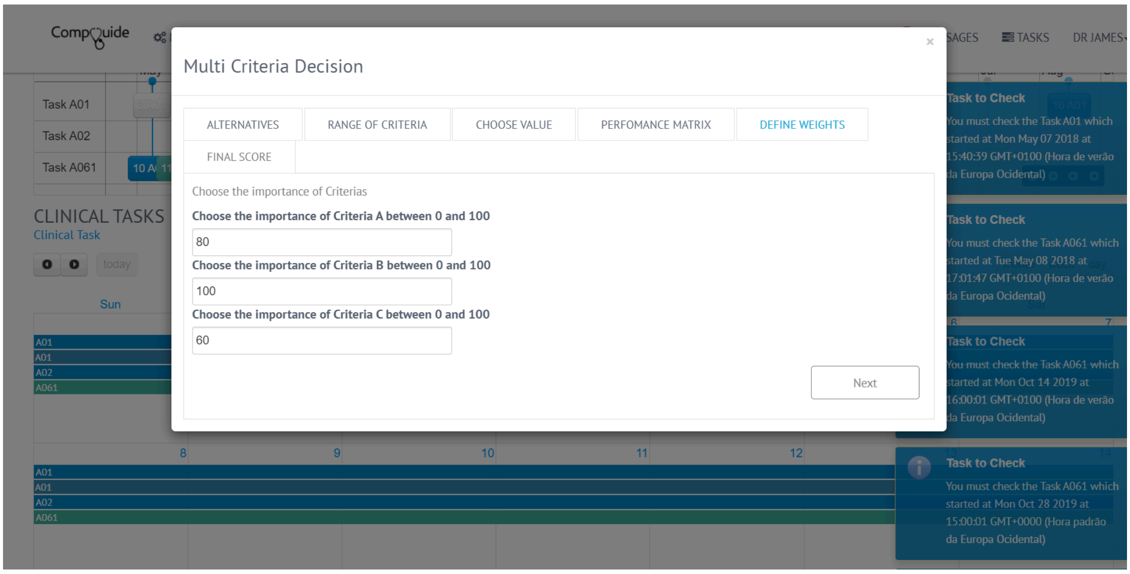
Then, Figure 19 shows the tab where the performance matrix of the before inputted values is displayed.



Criteria	Alternative 1	Alternative 2	Alternative 3
A - Severity of disease for which drugs advised	85	90	95
B - Adverse drug-drug interactions	70	70	75
C - Expected outcomes for the drug application	34	38	43

Figure 19: Performance matrix for conflict between task A02 and A61.

Afterwards, it is necessary to choose the importance of each criterion between the values 0 and 100, as shown in Figure 20.



Choose the importance of Criterias

Choose the importance of Criteria A between 0 and 100

80

Choose the importance of Criteria B between 0 and 100

100

Choose the importance of Criteria C between 0 and 100

60

Figure 20: Choice of importance of each criterion.

Finally, Figure 14 depicts an example of a table with the results of the MCDA process with the total and partial scores of each alternative, and which alternative is chosen by this process.

Appendix B

CompGuide Model for Mitigation of Recommendation Conflicts

This appendix provides details of CompGuide model for the representation of CPGs in a machine-interpretable format using OWL, the mechanism for identification of recommendation interactions, conflicts, and alternatives using existing terminology services (such as the RxNorm API), and the automated mechanism to reason over conflicting CIGs executed concurrently and solve conflicts with MCDA.

B.1 CompGuide Model for Determining Recommendation Interactions

In this section, we provide details on how the CompGuide model represents clinical recommendations from CPGs and their interactions. The System presented in this project can be used for knowledge representation in works such as [99]. Furthermore, we can use quality information metrics like the ones suggested in [100] to provide degree's of confidence for medical symptoms and findings.

B.1.1 Clinical Recommendations

The CompGuide model describes CPGs components in OWL. It represents CPGs in the form of a task network [101] and contains different types of clinical tasks. We only consider the *Action* task since it describes recommendations that should be carried out by the health care professional in daily clinical practice. This task consists of the following parameters:

- **Description:** the description of the action to be performed. (e.g. *Apply insulin 0.2 units/kg*)
- **Action type:** the type of action that should be performed by a health professional. It includes clinical procedures, clinical exams, medication recommendations and non-medication recommendations. For the purpose identifying interactions between actions, we will focus only on medication recommendations.
- **Outcomes:** A set of conditions that express the expected result of a task in terms of the changes produced in the patient's condition. The expected outcome is used as a criterion for assessing a corresponding *Action* within the proposed MCDA model (more details in B.2.3 *step 3*). It contains the following parameters:

  - **Value:** the value that quantifies the clinical parameter to be compared;
  - **Comparison operator:** includes the operators *equal_to*, *greater_than*, *greater_or_equal_than*, *less_than*, *less_or_equal_than* and *different_from*;
  - **Condition parameter:** the clinical parameter to be evaluated. (e.g. *temperature*, *glycemic level*);
  - **Unit:** the unit in which the parameter should be expressed in (e.g. *units/kg*);

- **Medication Recommendation:** We only address the Medication Recommendation action type (there are several other types in CompGuide). It concerns recommendations that advise drugs to treat diseases. It can be defined through the following parameters:
  - **Active Ingredient:** the chemical component of a drug responsible for the medication's effects;
  - **Dosage:** the drug dosage information;
  - **Pharmaceutical Form:** the information about the presentation of the drug (e.g. tablet, capsule, solution for injection, cream, etc.);
  - **Posology:** the information about the number of times it should be taken;
  - **Identifier:** the drug identifier.

B.1.2 Recommendation Interactions

We only focus on drug-drug interactions that have harmful effects on the patient's condition and also interactions that occur when an effect of one drug alters the effect of another co-administered drug. We use the RxNorm API [102] to identify conflicts (we provide the details on how to identify and mitigate the interactions in section B.2). RxNorm is a standard drug terminology that links different drug nomenclatures, and its interaction API uses two sources for its interaction information - ONCHigh and DrugBank. From this service, we extract the RxCUI (Concept Unique Identifier) for each drug, drug names, existing interactions and their severity, and the sources of relationships between drugs. Table 5 summarises the information extracted from RxNorm into our knowledge base.

Table 5: The information extracted from the RxNorm Interaction API.

Extracted Information	Description
RxCui	RxNorm drug identifier.
Severity	Severity of the interaction. N/A if the data source doesn't contain severity level information, and high if an adverse interaction exists.
Description	Description of the severity of the interaction.
Source Name	The data source that provides information about drug interactions. Currently, there are only two sources "DrugBank"and "ONCHigh".

In our model, interactions occur between different *Actions*. An *Interaction* results from a parallel execution of *Actions* of type Medication Recommendation. Based on data extracted from the RxNorm API (depicted in Table 5), the *Interaction* entity is defined by the following parameters:

- **Medication recommendation A:** a specific prescribed medication *A* that concerns an *Action A* (e.g. *Insulin*);
- **Medication recommendation B:** a specific prescribed medication *B* that concerns an *Action B* (e.g. *Goserelin*);
- **Severity:** the severity of the interaction that results for the concurrently application of *medication recommendation A* and *medication recommendation B*. It can assume the values described in Table 5;
- **Description:** the description of the interaction between *medication recommendation A* and *medication recommendation B* (e.g. *The terapeutic efficacy of Insulin can be decreased when used in combination with Goserelin*).

B.2 CIG Interaction Detection and Resolution

The present work provides a system that represents and identifies drug-drug interactions, using the RxNorm API and also provide alternative measures to mitigate these interactions. We use a mitigation function to calculate conflict-free alternative drugs. This function uses similarity between drugs, patient preferences over clinical recommendations, and clinician priorities over goals as mitigation principles.

The architecture is depicted in Figure 21. This is a three-level solution that encompasses the following stages of CIG deployment: representation of CIGs, identification of interactions, and generation of alternatives.

B.2.1 Representation of CIGs

To represent CIGs, we use the CompGuide ontology described in section B.1 and in more detail in [101]. The *CompGuide plugin* [98] provides step-by-step instructions to create and edit CIGs.

All encoded CIGs are stored in the *Guideline Repository*. This is a collection of owl files and can be accessed through the *Guideline Handler* which delivers the clinical tasks and respective constraints on demand to the Guideline Execution Engine (GEE).

B.2.2 Identification of Recommendation Interactions

The GEE performs verifications on task ordering, task constraints and task interactions by comparing the guideline care flow with the state of the patient and with concurrently executed guidelines. The result is a recommendation in the form of the next clinical task to be applied.

For the identification of task interactions, the GEE compares each concurrently executed *Actions* of all guidelines and their respective medication recommendations. For each pairwise combination of medication recommendations, the GEE calls the RxNorm Interaction API to retrieve the information in Table 5, as shown in Algorithm B.1.

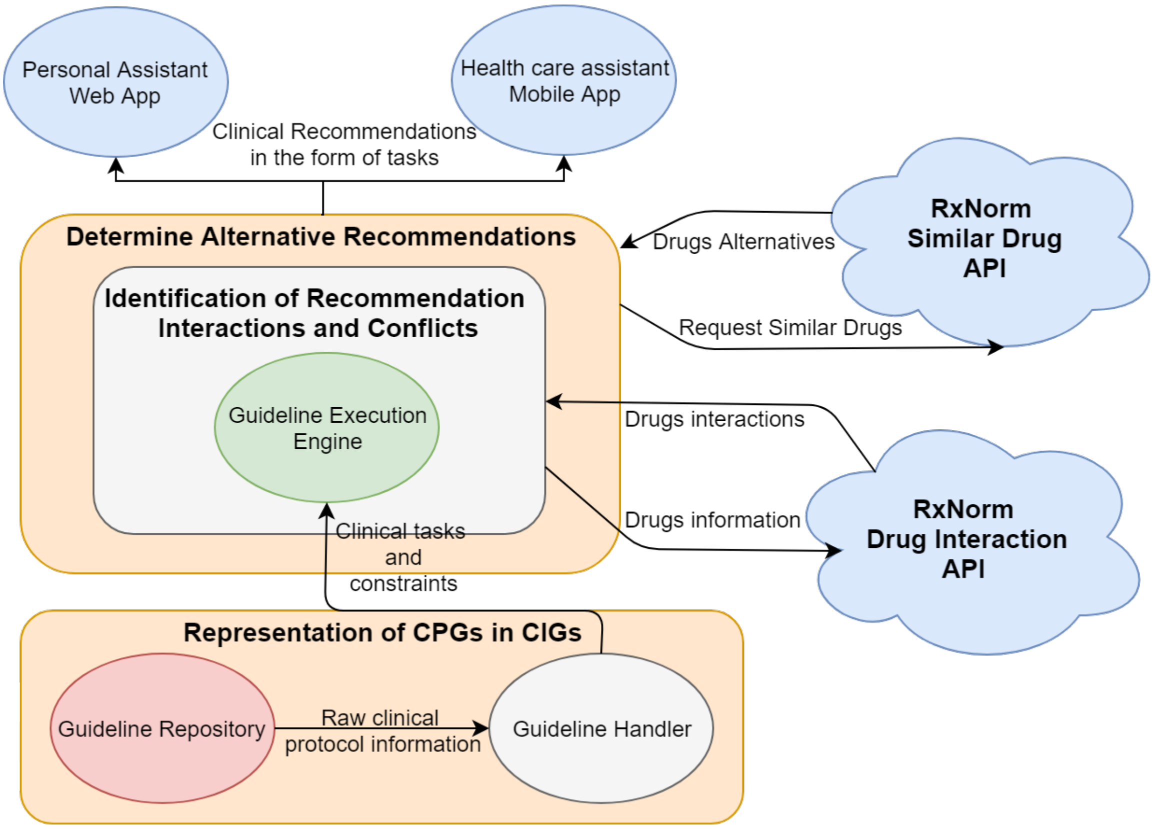

Personal Assistant Web App
Health care assistant Mobile App
Clinical Recommendations in the form of tasks
Determine Alternative Recommendations
Identification of Recommendation Interactions and Conflicts
Guideline Execution Engine
Drugs Alternatives
RxNorm Similar Drug API
Request Similar Drugs
Drugs interactions
Drugs information
RxNorm Drug Interaction API
Clinical tasks and constraints
Representation of CPGs in CIGs
Guideline Repository
Raw clinical protocol information
Guideline Handler

Figure 21: Architecture of CompGuide system.

Algorithm B.1 Find drug-drug Interactions using RxNorm Interaction API.

```
Data: $actionList$ - a list of executed actions.
Data: $medRec_A$ - a specific medication recommendation of an Action A.
Data: $medRec_B$ - a specific medication recommendation of an Action B.
Data: $int_{AB}$ - an interaction between Action A and Action B.

for $action_A \in actionList$ do  // Iterates the current list of recommendations to be executed
    $medRec_A \leftarrow getMedicationRecommendation(action_A)$
    if $medRec_A \neq null$ then
        for $action_B \in actionList$ do
            if $action_A \neq action_B$ then
                $medRec_B \leftarrow getMedicationRecommendation(action_B)$
                if $medRec_B \neq null$ then
                    $int_{AB} \leftarrow callRxNormInteractionAPI(medRec_A, medRec_B)$
                    $storeInteraction(int_{AB})$                // Store in database the interaction.
                end
            end
        end
    end
end
```

B.2.3 Generating Alternative Recommendations

After processing the interactions between concurrently executed clinical tasks, the GEE mitigates these interactions. The steps for the evaluation of alternatives is described below.

Step 1: Providing Alternative Recommendations within a Guideline

As can bee seen in Figure 22, the CompGuide model allows the definition of different task orderings, among which are alternative tasks. In the case of alternative tasks, they are executed instead of another as the result of an inference process guided by trigger conditions. When an alternative task (recommending alternative drugs) has an interaction with another task, the GEE reprocesses the trigger conditions to determine the conflict-free alternative task. Then, it gets all the medication recommendations of the alternative tasks and tries to find if drug-drug interactions exist in them, by calling the RxNorm Interaction API for each pairwise drugs of the task. After determining the alternative recommendations, they are made available through the Personal Assistant Web App and Healthcare assistant Mobile App. In these assistants, it is possible to visualise the clinical recommendations, currently being applied to patients, in a calendar widget and time axis. More details about these widgets can be seen in [103]. If there are no conflict-free alternative tasks, the system moves to *step 2*, described in the next section.

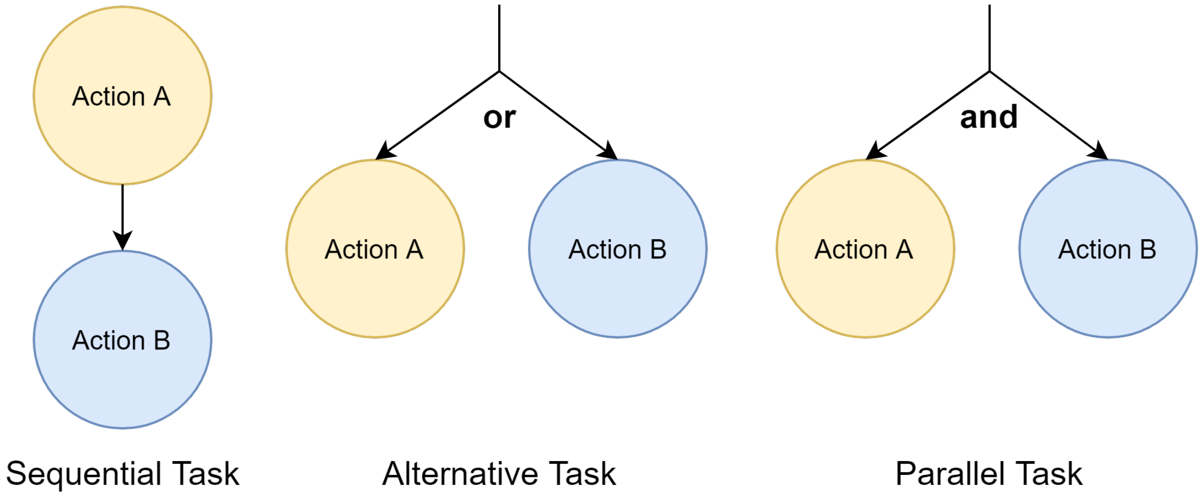


Figure 22: Types of tasks in CompGuide model.

Step 2: Providing Alternative Recommendations using RxNorm Similar Drugs API

In this step, the GEE provides alternative drugs using the RxNorm Similar Class API. For this purpose, it produces a ranking of alternative drugs based on the similarity score supplied by this service. Similar classes are defined as classes which have shared drug members with the drug members of the selected class. The similarity score provided by its services is a score that determines the similarity between drugs. From the RxNorm Similar Class API, we extract class name, class type, equivalence score and inclusion score. We only use the equivalence score for ranking the alternative drugs. The inclusion score is stored in our database to avoid the need for this information in future implementations. Table 6 summarises the information extracted to our knowledge base.

Thus, the GEE calls the RxNorm API to get alternative drugs for the given conflicted drugs and calculates the highest similarity score for the alternative medicines. For each alternative with a higher score, it tries to encounter a

Algorithm B.2 Find Alternative drugs using RxNorm similar drugs API.

```
Data: $ALTs_A$: the set of alternative recommendations of Action A
Data: $ALTs_B$: the set of alternative recommendations of Action B
Data: $ALT_A$: the alternative drug A with higher similarity score to drug A
Data: $ALT_B$: the alternative drug B with higher similarity score to drug B
Data: $Sim_A$: the similarity score of Drug A
Data: $Sim_B$: the similarity score of Drugs B
$ALT_A \leftarrow getAlternativeHighSimilarityScore(ALTs_A)$    // Get high score alternative drug
$ALT_B \leftarrow getAlternativeHighSimilarityScore(ALTs_B)$
$Sim_A \leftarrow similarityScore(ALT_A)$    // Get similarity score for each alternative drug
$Sim_B \leftarrow similarityScore(ALT_B)$
if $Sim_A > Sim_B$ then    // Determine which alternative will be used by comparing the similarity scores
    $INT_{ALT_A Drug_B} \leftarrow findInteraction(ALT_A, Drug_B)$    // Determine if there is a conflict between the drug and the alternative
    while $INT_{ALT_A Drug_B} == true\ OR\ ALT_A \neq null$ do    // If there is a conflict, try to find another alternative
        $ALT_A \leftarrow getNextAlternativeHighSimilarityScore(ALTs_A)$
        $INT_{ALT_A Drug_B} \leftarrow findInteraction(ALT_A, Drug_B)$
    end
    if $ALT_A \neq null$ then
        $storeAlternativeDrug(ALT_A)$    // Store in database the alternative
    end
else    // Alternative B was selected
    $INT_{ALT_B Drug_A} \leftarrow findInteraction(ALT_B, Drug_A)$
    while $INT_{ALT_B Drug_A} == true\ OR\ ALT_B \neq null$ do
        $ALT_B \leftarrow getNextAlternativeHighSimilarityScore(ALTs_B)$
        $INT_{ALT_B Drug_A} \leftarrow findInteraction(ALT_B, Drug_A)$
    end
    if $ALT_B \neq null$ then
        $storeAlternativeDrug(ALT_B)$
    end
end
```

conflict-free drug. If there is a conflict, the system finds the next alternative with the higher score, if there is no conflict, it stores the alternative in the knowledge base, as shown in Algorithm B.2. Then, the clinical recommendations are made available in before-mentioned assistant interfaces. If GEE does not find conflict-free alternative drugs, then it moves to the *step 3*, which will be described in the next section.

Table 6: The information extracted from RxNorm Similar Class API.

Extracted Information	Description
RxCui	The RxNorm identifier of the similar drug.
Class Name	The name of the class/drug.
Class ID	RxNorm class identifier. Used for future interactions with RxNorm services.
Equivalence Score	A score that measures the similarity between two classes.
Inclusion Score	It is a metric for finding specific classes that are included in broader classes.
Drug Source	The data source that provides information about the drug.

Step 3: Multiple Criteria decision Analysis For Clinical Mediation

The process of elicit stakeholder preferences on best decision alternatives and criteria should result from a discussion between the patient and the physician. The objective of MCDA here is to propose a ranking of solutions based on the before mentioned principles. To produce this ranking, we use an adaptation of Keeney and Raiffa MCDA approach [104], where for each alternative in given criteria, the decision makers establish the preferences within and between criteria, via scoring and weighting. Thus, it involves scoring using "partial value functions" and "swing" weighting. Once the decision problem is defined, the criteria and alternatives are as follows:

- **Alternative solutions:** they result from the combination of all conflicted recommendation medications currently being analysed. For instance, if we have a conflict between recommendation medication A and B, the solutions to be evaluated are:

  1. Application of recommendation medication A;
  2. Application of recommendation medication B; or
  3. Application of recommendation medications A and B.

- **Criterion:** the criterion regarding the type of problem are severity of disease for which drugs are advised, adverse drug-drug interactions, and expected outcomes for the drug application. The severity of disease for which drugs are advised is obtained from discussions between health care professionals and patients. It is not the objective of this doctoral work to specify how this discussion and specification takes place. The adverse drug-drug interactions are obtained through the RxNorm Interaction API. Outcomes are extracted from the *Outcome* data parameter in the *Action* class (section B.1.1).

The criteria expected outcomes for the drug application and severity of diseases for which drugs are advised are measured in units where higher performance is better, whereas, for criterion adverse drug-drug interactions, lower performance is better. The process of organising the performance of alternatives on each criterion help decision makers weigh up multiple criteria by highlighting the trade-offs that need to be made. We aggregate the performance in these criteria to produce an overall value. Thus, the objective is constructing and comparing numerical scores (total value) to identify the degree to which one decision alternative is preferred over another.

The scores on a criterion are expressed according to a measurement scale (e.g. 10-20 or 40-50). This allows the stakeholders to establish the importance of alternatives (in a range) in a given criterion. Then, we use functions to establish the relationship between the score on the criterion and its own MCDA score, since it is defined in a different scale (within a range between 0-100 points). This function should consider if the relationship between the variation along the defined scale is linear or not and if a high performance (e.g. treatment effectiveness) on criteria is better or if low performance (e.g. adverse drug events) on criteria is better. In the case of non-linear functions, we use the bisection method to produce partial value function.

The partial value function equation is as follows:

$$y = mx + b \tag{B.1}$$

Admitting the following choice of criteria range, range $R(x, y)$, it is possible to determine two points that belong to the partial value function P1(R_x, 0) and P2(R_y, 100), in case high performance in a criterion means a better score. In case low performance means better score, admitting range $R(x, y)$, the two points are P1(R_y, 0) and P2(R_x, 100). Thus, the *m* value is obtained by the following expression:

$$m = \frac{100 - 0}{P2_x - P1_x}$$

, where $P1_x$ and $P2_x$ are the values of R(x , y)

(B.2)

Next, we can obtain the *b* value using the following expression:

$$b = y - mx \tag{B.3}$$

Thus, admitting equation B.3, it is possible to replace *m* with formula B.2.3 and obtain the following:

$$b = y - (\frac{100 - 0}{P2_x - P1_x})x$$

, where $P1_x$ and $P2_x$ are the values of R(x , y)

(B.4)

Next step involves weighting criteria, which allows producing total values from partial value scores by applying weights. This is performed using swing weighting exercise, where it is assign 100 points to the criterion within a range of performance that matters most. Then, it is made a pairwise comparison between one criterion and the others, to assign the score (0 - 100 points) and determine the importance of swings in criteria. After, we normalise the points by dividing them by the sum of points. So, the weights of each criterion can be obtained by the following expression:

$$WeightC(i) = \frac{P^i}{\sum_{i=1}^{n} P^n} \tag{B.5}$$

where P^i denotes point i allocated to a specific criterion n.

After eliciting the scores and weights, we apply the aggregation method using the additive model to obtain total scores. The total score for each alternative is obtained by multiplying a numerical score for each option on a given criterion by the relative weight for the criterion and later summing these weighted scores. Thus, the total score is provided by the following expression:

$$f(n) = \sum_{n=1}^{n} S^n * WeightC^n \,,$$

where n is the number of solutions to be scored, S^n a score of a specific solution and $WeightC^n$ is the relative weight for a specific criterion.

(B.6)

Table 7 presents the additive model for the aggregation method. Thus, the total scores of each solution are made available through the Personal Assistant Web App and Healthcare assistant Mobile App, presenting the selected solution. Also, it is provided in these assistants the respective recommendation in the different widgets, namely time axis and calendar widget.

Table 7: Assessment of all possible solutions. The symbol *C* indicates a certain criterion to be evaluated for a given solution α. *S* means the score of the solution.

Solutions (α)	**Criterion** $(\boldsymbol{C})$			**Total Score**
	C^1	...	C^n	
α^1	S^1C^1	...	S^1C^n	$f(1) = \sum_{n=1}^{1} S^1 * WeightC^n$
...	...	...	...	...
α^n	S^nC^n	...	S^nC^n	$f(n) = \sum_{n=1}^{n} S^n * WeightC^n$

B.3 MCDA Case Example Implementation

In this section, we will implement a case study using the MCDA approach described in section B.2.3, more specifically the *step 3* of generating alternative recommendations, in case of conflicts and interactions between concurrently executed clinical recommendations. For this purpose, we used two CIGs based on the NCCN Clinical Practice Guideline for Prostate Cancer [31] and the IDF Clinical Practice Recommendations for managing Type 2 Diabetes [32], which were encoded in our ontology using the CompGuide plugin editor [98]. It is important to refer that in this case example, we do not process the temporal constraints.

In this case example, we will consider the following recommendations from the mentioned guidelines:

- *Recommendation A* belongs to the guideline for managing Type 2 Diabetes: "Apply insulin 0.2 units/kg and titrate once weekly at one unit each time during six months to achieve a target fasting blood glucose between 3.9 and 7.2 mmol/L (70 and 130 mg/dL)";

- *Recommendation B* belongs to the guideline for prostate cancer: "Apply leuprolide 180 mg/m2 as part of Androgen Deprivation Therapy".

As mentioned in section B.1.1, the clinical recommendations are mapped to the CompGuide ontology, being represented as depicted in Table 8. In this table, it is possible to visualise the values defined for the parameters as well as the different classes of the CompGuide model responsible for gathering the data (the details about these classes are provided in section B.1). These classes encompass *Action*, *Recommendation Medication* and *Outcome*.

Table 8: Instantiation of case example for *Recommendation A* and *Recommendation B*.

Recommendations	
Recommendation A "Apply insulin 0.2 units/kg and titrate once weekly at one unit each time during six months to achieve a target fasting blood glucose between 3.9 and 7.2 mmol/L (70 and 130 mg/dL)"	**Action** • Description: apply insulin • Action Type: medication recommendation **Outcome** • Value: 3.9 and 7.2 • Comparison operator: *greater_than* and *less_than* • Condition parameter: blood glucose • Unit: mmol/L **Medication Recommendation** • Active Ingredient: Insulin • Dosage: 0.2 units/Kg • Pharmaceutical Form: N/A • Posology: Insulin 0.2 units/kg given once weekly at one unit each time during six months • Identifier: N/A
Recommendation B "Apply leuprolide 180 mg/m2 as part of Androgen Deprivation Therapy"	**Action** • Description: apply leuprolide • Action Type: medication recommendation **Outcome** • Value: 50 • Comparison operator: *less_or_equal_than* • Condition parameter: serum testosterone • Unit: ng/dL **Medication Recommendation** • Active Ingredient: leuprolide • Dosage: 180 mg/m2 • Pharmaceutical Form: N/A • Posology: leuprolide 180 mg/m2 given • Identifier: N/A

In this case, the two recommendations are concurrently being applied and have drug conflicts, namely the drug leuprolide harms the therapeutic efficacy of insulin. This interaction was obtained by calling RxNorm Interaction API as described in section B.2.2. The information extracted to our knowledge base is summarised in Table 9. Once there is an interaction between recommendations, the application moves to *step 3* in order to produce a ranking of alternatives that can help stakeholders to choose the appropriate solution, as described in section B.2.3.

The criteria defined are as follows:

- **Criteria A**: Severity of disease for which drugs are advised;

Table 9: The information extracted from the RxNorm Interaction API for the given case example. The RxCui field is the output of the RxNorm API; thus it is in JSON format. The RxCui of insulin is 253182 and RxCui of leuprolide is 42375.

Extracted Information	Value
RxCui	”userinput”: [”insulin”: ”253182”, ”leuprolide”: ”42375”]
Severity	N/A
Description	The therapeutic efficacy of Insulin Human can be decreased when used in combination with Leuprolide.
Source Name	DrugBank

- **Criteria B**: Adverse drug-drug interactions;
- **Criteria C**: Expected outcomes for the drug application.

The corresponding alternatives for this case example are:

- **Alternative 1**: Only apply insulin 0.2 units/kg and titrate once weekly at one unit each time during six months to achieve a target fasting blood glucose between 3.9 and 7.2 mmol/L (70 and 130 mg/dL) and skip *alternative 2*;
- **Alternative 2**: Only apply leuprolide 180 mg/m2 as part of Androgen Deprivation Therapy and skip *alternative 1*;
- **Alternative 3**: Apply *alternative 1* and *alternative 2* simultaneously.

The result of scoring criteria is summarised in Table 10. It depicted the performance matrix for the given case example where stakeholders define the scores on each criterion. In this elicitation, they define the following ranges of each criterion: 80-100 aa for criterion A, 70-90 bb for criterion B and 30-50 cc for criterion C. The units aa, bb and cc are used to measure criteria A, B and C respectively.

Table 10: Performance matrix for the given case example.

Criteria	Alternative 1	Alternative 2	Alternative 3
A - Severity of disease for which drugs are advised	85 aa	90 aa	95 aa
B - Adverse drug-drug interactions	70 bb	70 bb	75 bb
C - Expected outcomes for the drug application	34 cc	38 cc	43 cc

Later, we developed the corresponding partial value functions to specify the relationship between the scoring scale and the score that will be the input to MCDA (we define the scale for MCDA between 0-100 points). This is a linear relationship, thus the partial value functions for Criteria A, B and C are shown in Figure 23, 24 and 25, respectively. For Criterion B, lower performance is better, whereas for Criteria A and C higher performance is better. For instance, the performance of Criteria A in alternative 1, which is 85 aa, corresponds in our linear partial function (Figure 24) to a score of 25 (in a scale of 0-100 points).

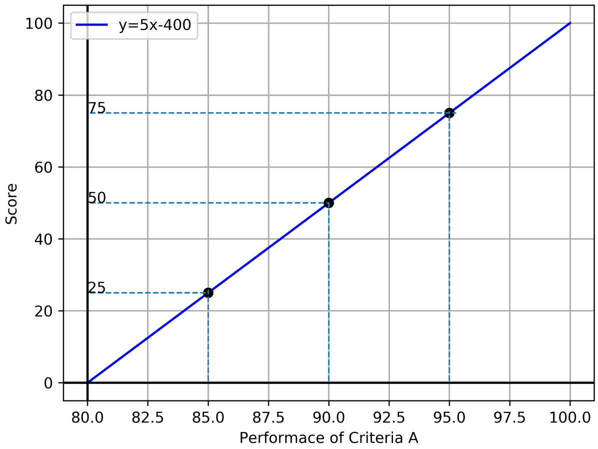


Figure 23: Linear partial function for Criteria A.

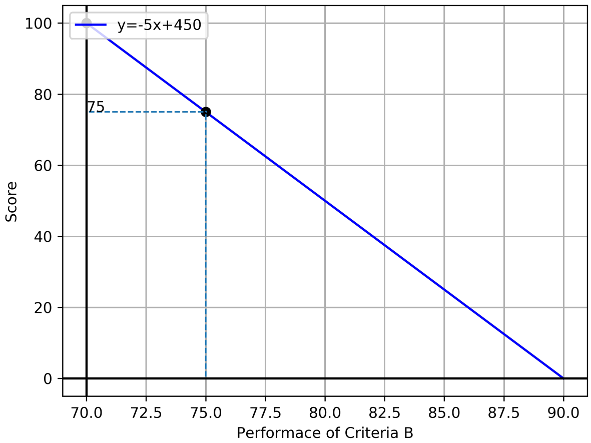


Figure 24: Linear partial function for Criteria B.

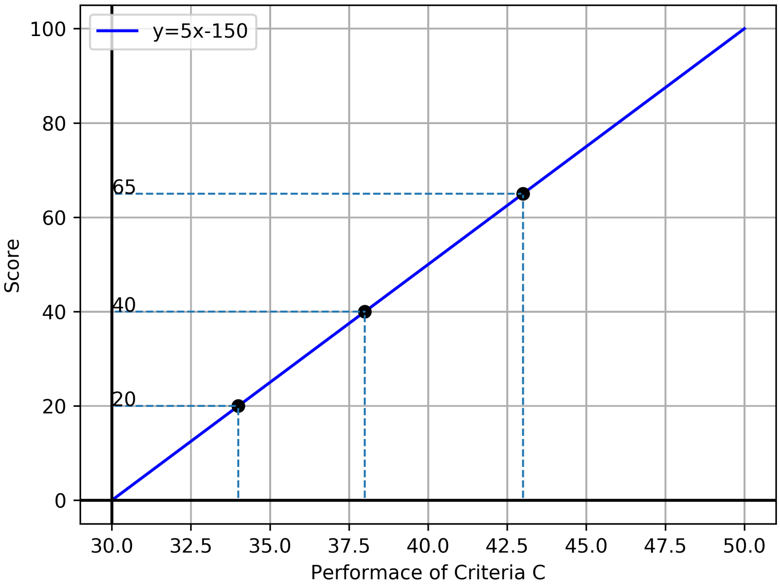
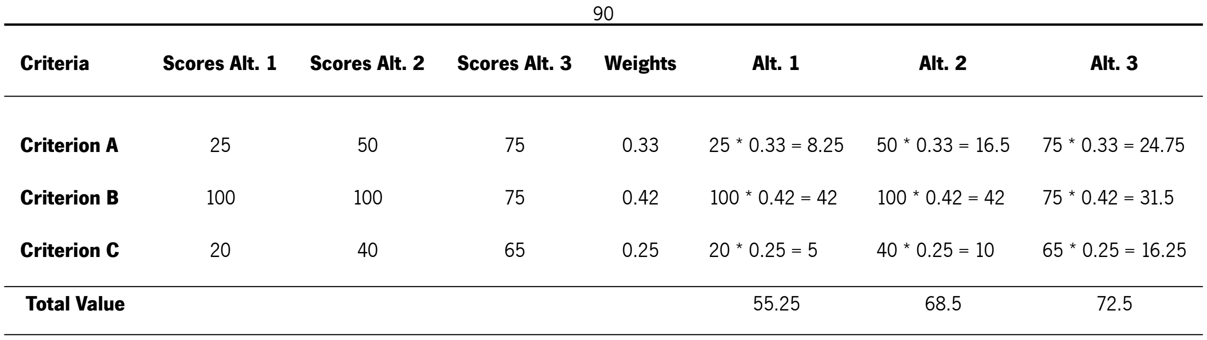


Figure 25: Linear partial function for Criteria C.

Then, we proceed with swing weighting exercise to obtain the weights of the criteria. This was accomplished by performing a pairwise comparison between one criterion and all the others to determine the relative importance of swings in criteria, and correspondingly allocate the points between 0 and 100. This exercise results in the following allocated points: Criteria A obtained 80 points; Criteria B obtained 100; and Criteria C obtained 60. After, we divided each criterion's points by the sum of points. Thus, the weights for the criteria are 0.33 for Criteria A, 0.42 for Criteria B and 0.25 for Criteria C. Finally, we produce the total values using the additive model. The scores, weights, and the total values are presented in Table 11. This provides a ranking of the alternatives, where alternative 3 (has the highest score, 72.5 points) is the preferred solution for this specific case followed by alternative 2 (68.5 points) and 3 (55.25 points).

Table 11: Aggregation method to produce the total values for each alternative.



90

Criteria	Scores Alt. 1	Scores Alt. 2	Scores Alt. 3	Weights	Alt. 1	Alt. 2	Alt. 3
Criterion A	25	50	75	0.33	25 * 0.33 = 8.25	50 * 0.33 = 16.5	75 * 0.33 = 24.75
Criterion B	100	100	75	0.42	100 * 0.42 = 42	100 * 0.42 = 42	75 * 0.42 = 31.5
Criterion C	20	40	65	0.25	20 * 0.25 = 5	40 * 0.25 = 10	65 * 0.25 = 16.25
Total Value					55.25	68.5	72.5