Automatically Identifying Drug Conflicts in Clinical Practice Guidelines

António Silva [1], Tiago Oliveira [2], Ken Satoh [2] and Paulo Novais[1]
[1] Algoritmi Centre, University of Minho, Braga, Portugal, asilva@algoritmi.uminho.pt, pjon@di.uminho.pt
[2] National Institute of Informatics, Tokyo, Japan, {toliveira,kstaoh}@nii.ac.jp

Abstract

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. However, when concurrently apply them, this can lead to adverse drug-drug interactions that can impair the patient's condition. Several efforts have been made in order to provide systems capable of identifying these conflicts. However, the current approaches for this purpose have some limitations. This paper presents a solution that represents CPGs as Computer-Interpretable Guidelines (CIGs) and allows for the automatic drug conflict identification and resolution. Also, we provide the identification of improvements to include in a future model. Moreover, this system provides clinical recommendations in an agenda, being capable of identifying drug interactions when drugs are prescribed simultaneously and provide conflict-free alternatives.

1 Introduction

To improve the utilisation of clinical practice guidelines (CPGs) at the point of care, there have been numerous efforts to computerise CPGs in ontologies and incorporate them within Clinical Decision Support Systems (CDSSs). There are several guideline description languages such as Arden Syntax (Samwald et al. 2012), Guideline Interchange Format (GLIF) (Peleg et al. 2000), Asbru (Balser, Duelli, and Reif 2002), EON (Musen et al. 1996), PROforma (Vier et al. 1997) and Guideline Acquisition, Representation and Execution (GLARE) (Bottrighi et al. 2006) that are aimed at the representation of CPGs as computer-interpretable guidelines (CIGs) in order to provide computer-assisted tools that help health care professionals make decisions. Through the formalisation of CIGs in CDSSs, a new range of operations can be performed with the knowledge they enclose. Such includes automated reasoning for the generation of recommendations, automatic identification of conflicts between different 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. So, the objective of these approaches is to provide support to the processes of diagnosis and planning of the clinical treatments, as well as to promote the use of the best clinical practices.

However, these systems lack the flexibility to support cases where multiple protocols need to be combined; this is especially problematic for patients with *multimorbidity*. *Multimorbid* patients have complex treatment plans and face a high burden of the disease since they suffer from multiple diseases at the same time. There are also several problems regarding the application of treatment plans of multiple disease-specific CPGs to multimorbid patients. Such includes adverse drug events, increased treatment complexity, and cost of treatment (Tinetti, Bogardus Jr, and Agostini 2004) (Boyd et al. 2005). Thus, the application of multiple CPGs individually can result in complex multiple drug regimens (polypharmacy) with the potential for harmful combinations of drugs (Dumbreck et al. 2015).

Therefore, new needs arise in order to provide computer-assisted tools that automatically identify the common potential conflicts or interactions that can happen when merging CIGs, namely, those that happen when there are drug-drug interactions.

The work described herein presents a system that automatically identifies recommendation interactions, conflicts, and alternatives using existing terminology services such as the RxNorm API. Thus, the contributions featured in this work are: characterisation of main approaches to handle the combination of CIGs, especially for *multimorbid* patients and a solution to address this problem.

The paper is organised as follows. Section 2 describes related work regarding systems for combining CPGs. Section 3 presents an architecture for combining CIGs as well as the contributions for the deployment of CPGs in CDSSs. Finally, Section 5 presents the conclusions drawn so far with the development of the system and future directions for the work.

2 Existing Systems for Identification of Conflicts Between Concurrently Executed CPGs

When treating *multimorbid* patients, health care professionals need to retrieve clinical recommendations from multiple chronic disease CPGs. From the combination of these recommendations, several problems can happen, for instance when a drug, prescribed for one condition, has an adverse effect on another condition (Boyd et al. 2005). With the growing number of *multimorbid* patients, identification of these inconsistencies becomes increasingly essential (Wilk et al.

2011). Computerised CDSSs have been used to alert health care professionals to adverse drug events at the point of care (De Clercq, Kaiser, and Hasman 2008). In this section, we will provide a literature review of the existing CDSSs for identification of conflicts between concurrently executed CIGs.

2.1 Constraint Logic Programming

Wilk et al. propose an approach that combines logic programming with constraint satisfaction problems (Wilk et al. 2011). They use CIGs as an activity graph and use constraint logic programming to identify and mitigate possible adverse interactions between CIGs, it means, to identify conflicts associated with potentially contradictory and adverse activities resulting from applying two CPGs to the same patient. Although this approach provides automatic identification of conflicts and solutions, it depends on the availability of knowledge bases containing information about both diseases and the whole work of combining CIGs remains manual. So, in order to provide automatic identification conflicts and solutions need to be defined in a *medical background knowledge* as protocol-dependent rules/constraints.

2.2 Rule-based Combinations

The RBC approach provides identification and reconciliation of drug conflicts between recommendations of two concurrently executed CPGs (López-Vallverdú, Riaño, and Collado 2013). 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, i.e., 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 pairs of CPGs, a final treatment plan based on two CPGs could again be combined in a pairwise manner with a new CPG.

2.3 OntoMorph

The objective of the *OntoMorph* (Jafarpour 2013) approach is to propose a treatment plan, consisting of several tasks, that do not conflict and that are time and resource efficient. Jafarpour et al. (Jafarpour and Abidi 2013) used ontologies to develop systems to merge two concurrent CPGs into a comorbid personalised guideline. They extracted clinical tasks from the CPGs and converted them to CIGs 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 XML file (Jafarpour and Abidi 2013). 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 in 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).

They also built 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 problems were not yet (completely) 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 (Zamborlini et al. 2016). 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 Architecture to Automatically Identify Drug Interactions and Conflicts

There are some limitations regarding the systems aforementioned which should be taken into account. These limitations regard with the necessity of manually defining all guideline-dependent rules, limitations in the number of CIGs that can be combined and necessity of all conflicts and solutions to be available in a knowledge base.

Thus, 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 CPGs. To accomplish the goal mentioned above, as shown in Figure 1, we provide a solution in three levels: 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 provide an explanation of the architecture in the different stages regarding the different level mentioned before.

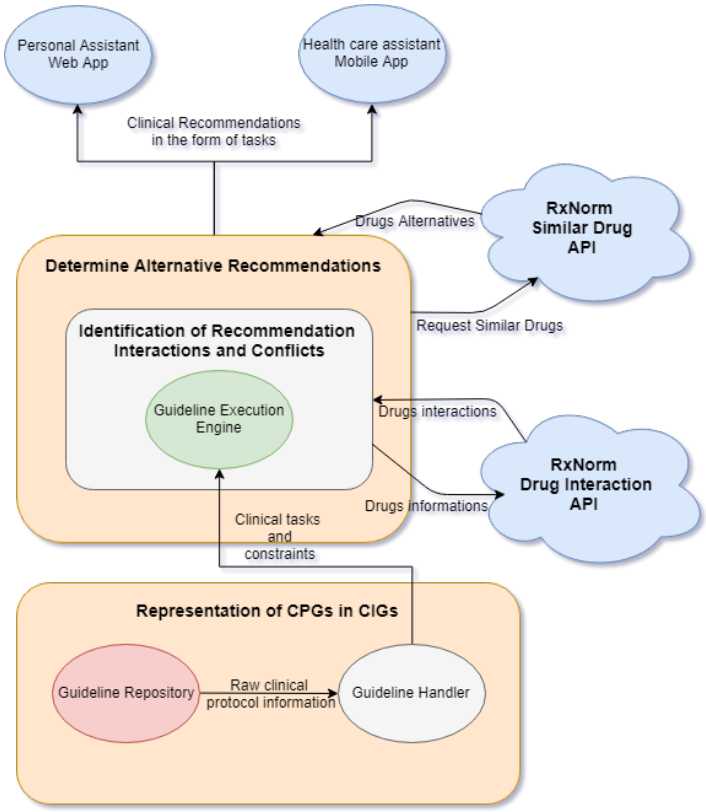

Figure 1: Architecture of CompGuide system

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 (Oliveira, Novais, and Neves 2013) contains different types of clinical tasks such as *Question*, *Action*, *Decision*, *End*, *Plan* and *Condition* and constraints expressed in the form of conditions on the patients state, such as *TriggerConditions*, *PreConditions* and *Outcomes*. Moreover, it provides a model of temporal representation (Oliveira et al. 2017) 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 (Gonçalves et al. 2017). This plugin performs the role of managing the creation and editing of CIGs.

The final output will be a CIG that will be saved in the *Guideline Repository*. This component is responsible for keeping different CIGs represented according to the CompGuide ontology. The *Guideline Handler* is responsible for managing the access to recommendations of CIGs 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 Execution Engine* with the information of the clinical tasks provided by the *Guideline Handler* 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 times of the tasks.

This component is also responsible for calling the RxNorm service in order to identify the interactions and recommendation conflicts. RxNorm (Liu et al. 2005) integrates the Unified Medical Language System and offers normalised names for clinical drugs and links its names to many of the drug vocabularies commonly used in pharmacy management and drug interaction software, including those of First Databank, Micromedex, MediSpan, Gold Standard Drug Database, and Multum. The RxNorm interaction API uses two sources for its interaction information - ONCHigh and DrugBank. The RxNorm interaction API provides 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 Determine Alternative Recommendations

With the severity of recommendation interactions, it is possible to determine the necessity of providing alternative recommendations. The severity can assume values such as *high* if there is an adverse drug event resulting from the interaction, and *N/A*, if there is no adverse effect. In case the interaction between drugs assumes as value *high* we call RxNorm RxClass API. This service provides alternative recommendations by offering ways to get similar classes of drug members. 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.

After processing the constraints of clinical tasks, determining the interactions between drugs and their alternatives, the clinical recommendations are made available through the Personal Assistant Web App and the health care assistant Mobile App.

The Personal Assistant Web Application access the data through the web services available in the CompGuide system (Silva et al. 2017). This component 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.

4 Execution 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 (Mohler et al. 2018) and the IDF Clinical Practice Recommendations for managing Type 2 Diabetes (Aschner 2017). These guidelines were a comprehensive case study since it was possible to test several aspects of the deployment of CIGs, namely those that concern with CIG representation, acquisition and execution in the CompGuide system. So, it is possible to examine all the stages of the deployment of CIGs, representing several types of tasks, various temporal constraints and several conflicts among the guidelines. However, in this section, we only address the conflicts between recommendations from many guidelines.

For demonstrations 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".

The recommendation A has the action 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. 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.

In this case, the two recommendations are concurrently being applied in 18th of July of 2018 and have drug conflicts, namely the drugs goserelin, leuprolide, histrelin and triptorelin have adverse effects on the therapeutic efficacy of insulin. The Figure 2 shows the output regarding these conflicts.

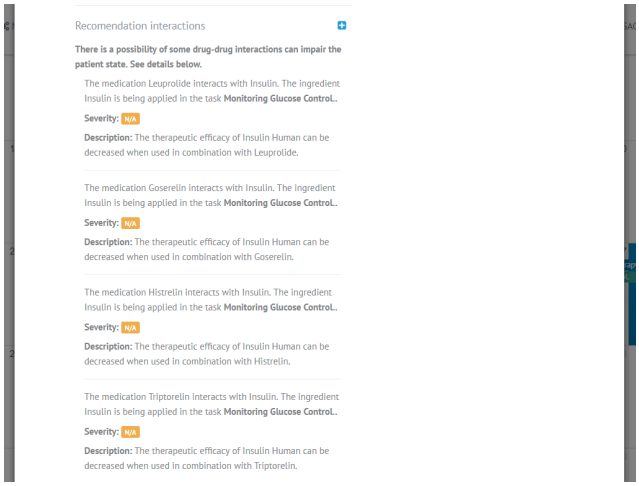

Figure 2: Recommendation interactions between recommendations A and B in the CompGuide Personal Assistant Web Application.

Later, the application tries to provide alternative drugs to address the identified conflicts, by calling RxNorm API as described in section 3.3. Through a mitigation function, the system calculates which one will be applied. This function has different mitigation principles, such as the similarity between drugs or user preferences. The objective is to determine which alternatives best fit the needs of users. One possible principle, which possibly will increase the effectiveness of this function, is a multiple criteria mechanism for supporting decision-making such as Multiple-criteria Decision Analysis (MCDA) (Thokala et al. 2016). This method allow to evaluate possible solutions based on conflicting criteria in decision problems. There are complex drug interactions that can impair the patient's condition as well as several solutions with conflicting objectives. Thus, it is essential to evaluate the possible solutions according to criteria such as user preferences in the best treatment alternative, benefit/risk assessment of different decision alternatives, the similarity between different drugs, the severity of disease for which recommendations are advised, among others. However, in this case study we only use the similarity between drugs as the mitigation principle. This function is responsible for finding the conflicts between drugs. For each conflict this function finds alternative drugs by calling the RxNorm API, according to section 3.3. Later, it calculates the high similarity score provided by RxNorm API for the set of alternatives drugs. 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 and displays the alternative drug. Based on the given case example, the reproduced recommendation alternatives are shown in Figure 3.

In the work described herein, we provide a system that automatically identifies conflicts and interactions between drugs for many guidelines. Comparing with *OntoMorph*,

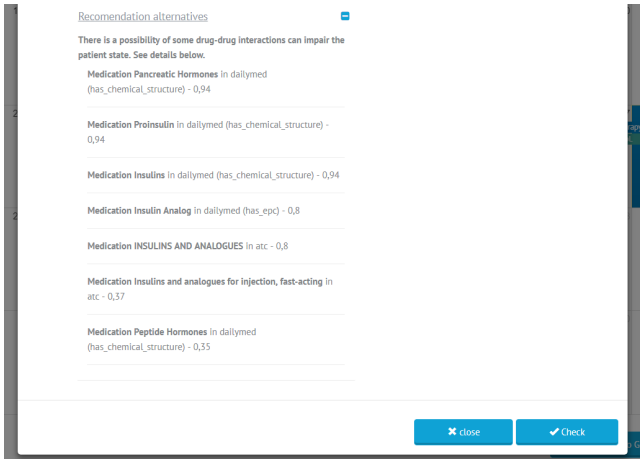

Figure 3: Recommendation alternatives for the given case example in the CompGuide Personal Assistant Web Application.

CLP and RBC, 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. So, using existing terminology services is one of the possible solutions for the limitation mentioned above. Other solution in regard to the usage of meta-rules, such 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, an MCDA approach can be a possible solution, since it allows the evaluation of possible solutions based on conflicting criteria. In the given example we use a mitigation function to determine which alternatives best fit the needs of users, by using as a mitigation principle, the similarity between drugs. One possible principle is a multiple criteria mechanism for supporting decision making such as Multiple-criteria Decision Analysis (MCDA) (Thokala et al. 2016). 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, the severity of disease for which recommendations are advised, benefit/risk analysis, and so forth.

5 Conclusions and Future Work

There are several efforts in order to provide systems capable of determining drug-drug interactions and conflicts among guidelines. However, some of the studied systems 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.

Although we currently do not provide an MCDA approach, it is our intention to implement 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 since some treatment plans can have harmful effects on the patients health . 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. Also, 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.

6 Acknowledgments

This work has been supported by COMPETE: POCI-01-0145-FEDER-0070 43 and FCT Fundação para a Ciência e Tecnologia within the Project Scope UID/CEC/00319/2013. The work of Tiago Oliveira was supported by JSPS KAKENHI Grant Number JP18K18115.

References

Aschner, P. 2017. New idf clinical practice recommendations for managing type 2 diabetes in primary care.

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

Bottrighi, A.; Terenziani, P.; Montani, S.; Torchio, M.; and Molino, G. 2006. Clinical guidelines contextualization in GLARE. In *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, volume 2006, 860. American Medical Informatics Association.

Boyd, C. M.; Darer, J.; Boult, C.; Fried, L. P.; Boult, L.; and Wu, A. W. 2005. Clinical practice guidelines and quality of care for older patients with multiple comorbid diseases: implications for pay for performance. *Jama* 294(6):716–724.

De Clercq, P.; Kaiser, K.; and Hasman, A. 2008. Computer-interpretable guideline formalisms. *Studies in health technology and informatics* 139:22.

Dumbreck, S.; Flynn, A.; Nairn, M.; Wilson, M.; Treweek, S.; Mercer, S. W.; Alderson, P.; Thompson, A.; Payne, K.; and Guthrie, B. 2015. Drug-disease and drug-drug interactions: systematic examination of recommendations in 12 UK national clinical guidelines. *Bmj* 350(mar11 2):h949–h949.

Gonçalves, F.; Oliveira, T.; Neves, J.; and Novais, P. 2017. Compguide: Acquisition and editing of computer-

interpretable guidelines. In *World Conference on Information Systems and Technologies*, 257–266. Springer.

Jafarpour, B., and Abidi, S. S. R. 2013. Merging disease-specific clinical guidelines to handle comorbidities in a clinical decision support setting. In *Conference on Artificial Intelligence in Medicine in Europe*, 28–32. Springer.

Jafarpour, B. 2013. Ontology merging using semantically-defined merge criteria and owl reasoning services: towards execution-time merging of multiple clinical workflows to handle comorbidity. *Dalhousie University*.

Liu, S.; Ma, W.; Moore, R.; Ganesan, V.; and Nelson, S. 2005. Rxnorm: prescription for electronic drug information exchange. *IT professional* 7(5):17–23.

López-Vallverdú, J. A.; Riaño, D.; and Collado, A. 2013. Rule-based combination of comorbid treatments for chronic diseases applied to hypertension, diabetes mellitus and heart failure. In *Process Support and Knowledge Representation in Health Care*. Springer. 30–41.

Mohler, J. L.; Lee, R. T.; Antonarakis, E. S.; Armstrong, A. J.; D'Amico, A. V.; Davis, B. J.; Dorf, T.; Eastham, J. A.; Ellis, R.; Enke, C. A.; and Farrington, T. A. 2018. National Comprehensive Cancer Network - Prostate Cancer. Technical report, National Comprehensive Cancer Network.

Musen, M. A.; Tu, S. W.; Das, A. K.; and Shahar, Y. 1996. EON: A Component-Based Approach to Automation of Protocol-Directed Therapy. *Emerging Infectious Diseases* 3(6):367–388.

Oliveira, T.; Silva, A.; Neves, J.; and Novais, P. 2017. Decision support provided by a temporally oriented health care assistant. *Journal of medical systems* 41(1):13.

Oliveira, T.; Novais, P.; and Neves, J. 2013. Representation of clinical practice guideline components in owl. In *Trends in Practical Applications of Agents and Multiagent Systems*. Springer. 77–85.

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

Samwald, M.; Fehre, K.; de Bruin, J.; and Adlassnig, K. P. 2012. The Arden Syntax standard for clinical decision support: Experiences and directions. *Journal of Biomedical Informatics* 45(4):711–718.

Silva, A.; Oliveira, T.; Neves, J.; Satoh, K.; and Novais, P. 2017. A system for the management of clinical tasks throughout the clinical process with notification features. In *Agents and Multi-Agent Systems for Health Care*. Springer. 76–93.

Thokala, P.; Devlin, N.; Marsh, K.; Baltussen, R.; Boysen, M.; Kalo, Z.; Longrenn, T.; Mussen, F.; Peacock, S.; Watkins, J.; et al. 2016. Multiple criteria decision analysis for health care decision makingan introduction: report 1 of the ispor mcda emerging good practices task force. *Value in health* 19(1):1–13.

Tinetti, M. E.; Bogardus Jr, S. T.; and Agostini, J. V. 2004. Potential pitfalls of disease-specific guidelines for patients with multiple conditions. *The New England journal of medicine* 351(27):2870.

Vier, E.; Fox, J.; Johns, N.; Lyons, C.; Rahmanzadeh, A.; and Wilson, P. 1997. PROforma: systems. *Computer Methods and Programs in Biomedicine* 2607(97).

Wilk, S.; Michalowski, M.; Michalowski, W.; Hing, M. M.; and Farion, K. 2011. Reconciling pairs of concurrently used clinical practice guidelines using constraint logic programming. In *AMIA Annual Symposium Proceedings*, volume 2011, 944. American Medical Informatics Association.

Zamborlini, V.; Hoekstra, R.; Da Silveira, M.; Pruski, C.; Ten Teije, A.; and Van Harmelen, F. 2016. Inferring recommendation interactions in clinical guidelines. *Semantic Web* 7(4):421–446.