Journal of Biomedical Informatics 46 (2013) 341–353

Contents lists available at SciVerse ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Mitigation of adverse interactions in pairs of clinical practice guidelines using constraint logic programming

Szymon Wilk [a,*], Wojtek Michalowski [b], Martin Michalowski [c], Ken Farion [d], Marisela Mainegra Hing [b], Subhra Mohapatra [e]

[a] Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
[b] Telfer School of Management, University of Ottawa, 55 Laurier Ave. East, Ottawa, ON, Canada K1N 6N5
[c] Adventium Labs, 111 Third Ave. South, Suite 100, Minneapolis, MN 55401, USA
[d] Division of Emergency Medicine, Children's Hospital of Eastern Ontario, 401 Smyth Rd., Ottawa, ON, Canada K1H 8L1
[e] Faculty of Medicine, University of Ottawa, 451 Smyth Rd., Ottawa, ON, Canada K1H 8M5

ARTICLE INFO

Article history:
Received 9 October 2012
Accepted 10 January 2013
Available online 30 January 2013

Keywords:
Clinical practice guideline
Comorbid diseases
Constraint logic programming
Domain knowledge
Clinical decision support

ABSTRACT

We propose a new method to mitigate (identify and address) adverse interactions (drug–drug or drug–disease) that occur when a patient with comorbid diseases is managed according to two concurrently applied clinical practice guidelines (CPGs). A lack of methods to facilitate the concurrent application of CPGs severely limits their use in clinical practice and the development of such methods is one of the grand challenges for clinical decision support. The proposed method responds to this challenge.

We introduce and formally define logical models of CPGs and other related concepts, and develop the mitigation algorithm that operates on these concepts. In the algorithm we combine domain knowledge encoded as interaction and revision operators using the constraint logic programming (CLP) paradigm. The operators characterize adverse interactions and describe revisions to logical models required to address these interactions, while CLP allows us to efficiently solve the logical models – a solution represents a feasible therapy that may be safely applied to a patient.

The mitigation algorithm accepts two CPGs and available (likely incomplete) patient information. It reports whether mitigation has been successful or not, and on success it gives a feasible therapy and points at identified interactions (if any) together with the revisions that address them. Thus, we consider the mitigation algorithm as an alerting tool to support a physician in the concurrent application of CPGs that can be implemented as a component of a clinical decision support system. We illustrate our method in the context of two clinical scenarios involving a patient with duodenal ulcer who experiences an episode of transient ischemic attack.

1. Introduction

A *clinical practice guideline* (CPG) [1] is a knowledge-based tool for disease-specific patient management. It is created by a panel of experts and is supported by medical evidence coming from specialized repositories [2] (such as The Cochrane Library [3]). A CPG encapsulates best practices in identifying relevant patient data, drawing conclusions from this data with regards to possible diagnoses, and prescribing the most appropriate treatment. It is generally agreed that the use of CPGs at the point of care has a positive impact on a patient's outcomes and providing guidelines to physicians in a computer executable format advances their adherence to standards of practice and improves quality of care [4]. While the ability to enact a CPG for available patient data plays an important role in guideline uptake by the medical profession, it does not address the issue of using guidelines on patients with comorbid diseases. Such a need is especially clear for elderly patients – several studies [2,5,6] have shown that about half of people 65 years or older have two or more comorbid conditions. The concurrent application of disease-specific CPGs for such patients often has undesired consequences due to the inconsistencies in the guidelines that lead to adverse drug–drug or drug–disease interactions [7].

Providing methods and tools that would enable the application of multiple CPGs to a patient with comorbid diseases has been listed as one of the "grand challenges" for clinical decision support [8]. The authors maintain that CPGs have been underutilized in practice because they have not accounted for co-morbidity issues. At the same time the authors call for new "combinatorial, logical, or semantic approaches" that would allow for merging several CPGs and cross-checking their recommendations. The call for

* Corresponding author. Fax: +48 61 877 15 25.
E-mail address: szymon.wilk@cs.put.poznan.pl (S. Wilk).

1532-0464/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jbi.2013.01.002

formal methods and theories for advanced analysis of CPGs (e.g., checking for inconsistencies and adverse interactions) has been repeated in [9].

The method proposed in this paper responds to these calls. It builds on our earlier research [10,11] that applied constraint logic programming (CLP) [12] to identify adverse interactions resulting from using multiple guidelines on the same patient. However, current research significantly expands our past work by:

- Introducing logical models of analyzed CPGs that act as an intermediate layer between common CPG representations and CLP models.
- Introducing interaction and revision operators that codify domain knowledge indicating possible adverse interactions between logical models and describing necessary revisions of the models.
- Proposing a new algorithm (further referred to as the *mitigation algorithm*) that operates on the logical models and uses operators in order to identify and address adverse interactions that are demonstrated by the concept of a potential source of infeasibility.

From a clinician's perspective we see our method (the mitigation algorithm in particular) as an "alerting tool" to support the physician in the concurrent application of multiple CPGs by warning about possible adverse interactions that may occur and by suggesting a clinically appropriate resolution of these interactions.

In order to ground our approach, we make the following simplifying assumptions:

- The number of mitigated CPGs is limited to two due to computational and logical complexities associated with the identification and resolution of adverse interactions.
- A CPG does not include parallel actions that can be followed concurrently (however, they can be replaced by a single sequence of actions).
- Temporal aspects of CPG modeling and execution are not taken into account (we only consider situations where guidelines are applied during a single patient-physician encounter).
- There is full semantic interoperability between the CPGs.
- A CPG is revised so it does not contain sub-guidelines (if sub-guidelines are present, they are expanded and included in the main guideline).

In Section 5 we describe how some of the above assumptions might be relaxed. Additionally, we assume individual CPGs are logically coherent, thus it is always possible to derive a feasible therapy if a single guideline is applied to a patient. This assumption is consistent with the intended purpose of a CPG in practice (i.e., it is designed to always propose a therapy for a given patient).

The paper is organized as follows. We start with related work. This is followed by a description of methodologies, including a brief presentation of the CLP paradigm, and a detailed description of our method. Then we illustrate our proposal using a simplified clinical case study. We conclude with a discussion and present areas for future research.

2. Related work

Related research on advanced analysis and processing CPGs falls under the following categories: (1) verifying a CPG [13–15], (2) using a CPG to critique actual actions [16,17], and (3) combing multiple CPGs for a patient with comorbid diseases [18–22]. The common theme shared by all categories is expressing processed CPGs using one of the computer-interpretable representations (e.g., GLIF3, SAGE, Asbru or PROforma, see [23–25] for a review) or ontological modeling.

The goal of CPG verification is early identification (before a guideline is applied to a patient) of issues such as semantic errors and inconsistencies in the definition of a guideline. Three types of methods are employed for this purpose: theorem proving [14], model checking [13] and knowledge-based checking [15]. Theorem proving implies representing a CPG as a theory that can be processed by an automatic theorem prover, and a failed proof indicates problems with the CPG. The use of theorem proving is described for example in [14], where the authors proposed a translation of Asbru constructs into a format acceptable by the KIV prover and applied it in order to verify two CPGs. Model checking methods assume checking a formal model of the CPG (usually obtained from its computer-interpretable representation) against a set of specifications describing expected properties of the guideline model. Specifications that cannot be verified indicate problems in the CPG. For example, in [13] the authors proposed a framework where a CPG represented as an UML state chart is being checked against a set of common types of the requirements in the form of temporal logical statements. Finally, knowledge-based checking methods employ methodologies developed for rule-based systems, where a CPG is represented as a set of rules that are being verified for possible anomalies (e.g., semantic errors). Definitions of anomalies can be either generic or domain-specific and are derived using expert knowledge. Such a knowledge-based checking method is described in [15], where the authors identify anomalies as unsatisfiable conditions, unreachable sequence of states or ambiguous state transitions.

Research on critiquing looks at comparing actual actions performed by a clinician to "ideal" actions suggested by a CPG in order to identify various types of non-compliance (e.g., conflicting actions). While conceptually different from CPG verification (critiquing is used when a CPG is being applied to a patient), it often uses similar methods, mostly model checking. In [16] the authors proposed an approach where a CPG represented as a state transition model is being checked against a set of temporal logic formulas describing actual actions. The proposed approach is able to identify two classes of non-compliance – related to actions (e.g., conflicting or missing mandatory actions) and related to clinical findings or patient data (e.g., missing relevant findings or wrong finding for a given action). A different approach to critiquing is presented in [17], where a CPG is translated into a set of "if conditions, then criticize" rules that are further used to verify treatments prescribed by physicians, and to provide additional explanation why a particular treatment should not be recommended.

It is worth mentioning that there are approaches combining CPG verification with compliance assessment for more comprehensive decision support. For example, the approach described in [26] uses an ontological model of domain knowledge together with abductive reasoning to evaluate whether a CPG is complete and appropriate (in terms of suggested actions and their justification), and to assess the compliance of a physician's actions with a CPG. Moreover, it provides support when executing a CPG by summarizing and explaining observed patient's clinical states and by warning about the violation of so-called safety rules. These rules are part of the ontological model and characterize adverse drug–drug or disease–drug interactions; they may also explain how specific interactions can be prevented.

Despite its clinical importance, research on combining multiple CPGs is in its early stages – as stated in [18], combining multiple CPGs "is not a trivial task". Proposed solutions vary from manual interventions, where human experts combine CPGs using a specialized editing tool [19], through semi-automatic approaches, where experts resolve automatically discovered conflicts [18], to fully automatic approaches that employ codified expert knowledge