Decision Support Systems 55 (2013) 499–515

Contents lists available at SciVerse ScienceDirect

Decision Support Systems

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

CONFlexFlow: Integrating Flexible clinical pathways into clinical decision support systems using context and rules

Wen Yao [a], Akhil Kumar [b,*]

[a] College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802, USA
[b] Smeal College of Business, The Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Available online 6 October 2012

Keywords:
CDSS
Clinical pathway
Rules
Context
Ontologies
Ad hoc subprocess

ABSTRACT

We propose CONFlexFlow (Clinical cONtext based Flexible workFlow) as a novel approach for integrating clinical pathways into Clinical Decision Support Systems (CDSS). It recognizes that clinical pathways involve frequent deviations and require considerable flexibility. Thus, integration of flexible pathways is critical for the success of CDSS. Further, it is based on a better understanding of clinical context through ontologies, and bringing them to bear in deciding the right rules for a certain activity. We also describe an approach for dynamically realizing context dependent medical activities in a clinical pathway based on the needs of a specific case. To illustrate the feasibility of our approach, we propose an implementation framework and present a proof of concept prototype using multiple open source tools. The role of semantic web technologies in realizing flexible clinical pathways and integrating them into CDSS is highlighted. Preliminary results from our initial implementation are discussed.

1. Introduction

Workflows play an important role in clinical environments by delineating the steps through which the treatment of a patient progresses. The temporal order and correct coordination of the various steps are clearly important. The workflows in these environments are called *clinical pathways*. These pathways generally follow well-established standards or clinical guidelines. However, they differ from other workflows found in business and production environments because clinical processes involve frequent deviations, and hence there is a need for considerable flexibility. Typically, a medical facility develops clinical pathways from clinical guidelines on the basis of its local resources and settings. Moreover, the pathway is further customized into a treatment scheme to suit an individual patient's needs [3].

Clinical workflows are highly dynamic, context sensitive, event driven, and knowledge intensive. In this respect, they are quite unique. In general, a patient interacts with a Primary Care Physician's (PCP) office, a pharmacy, labs, and one or more specialists, etc. In this setting, it is important to maintain coordination and flow of information among these various entities to ensure an optimal outcome. The need for new modeling techniques for designing such flexible workflows is motivated by several considerations. First, although many research efforts are geared towards establishing international healthcare standards (e.g., HL7 [22]) and a representation for sharable guidelines (e.g., GLIF [36]) for clinical practice, formal models of executable and flexible clinical workflows are very few (e.g., [13,55]). The execution of a clinical workflow is highly dependent on the existing body of medical knowledge, available resources, and specific case data. For example, doctors with different skill levels and fields of expertise may offer differing treatments to the same patient. A sudden rise in a patient's blood pressure may require an additional test and alter her treatment in the subsequent pathway. Thus, different pathways can arise based on case specifics and the proclivities of attending doctors, and it is important to formally model these scenarios. Second, since medical staff handles a lot of cases each day, they are prone to making mistakes in prescribing medications, performing procedures, and even making diagnoses [8,29,51]. Hence, Clinical Decision Support Systems (CDSS) and Computer Interpretable Guidelines (CIGs) [49] can assist care professionals in reducing the likelihood of errors and improving care quality.

Our goal is to show how flexible clinical pathways can be designed taking into account medical knowledge in the form of rules, and also detailed contextual information, for a medical workflow involving multiple participants to improve care quality. We propose a methodology for designing formal workflow models that capture medical knowledge and context in a common framework, and yet allow flexibility. Since a clinical workflow should naturally be aligned with clinical guidelines, it is necessary to ensure that it is formal and correct so that integrating decision support into this workflow can be helpful. The methodology is based on a formal rule and context taxonomy using ontologies. The rule taxonomy organizes the rules into a hierarchy while context encompasses aspects of patients, providers, resources, and environment. We will focus on how context is captured, described and summarized, and how rules are developed. A proof of concept prototype is built and

* Corresponding author. Tel.: +1 814 863 0034; fax: +1 814 863 7067.
E-mail addresses: yao.wen85@gmail.com (W. Yao), akhil@psu.edu (A. Kumar).

0167-9236/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2012.10.008

preliminary results are given to show the feasibility of our approach. In this paper, the terms pathway and workflow are used interchangeably.

The organization of this paper is as follows. Section 2 provides background and related work. Then we introduce a meta-model for our CONFlexFlow system and present the architecture for system implementation in Section 3. Next, Section 4 presents an ontology-based context model and discusses rule-based medical reasoning. Based on this knowledge framework, we present our approach for designing flexible clinical workflows using BPMN 2.0 ad hoc subprocesses and describe a preliminary implementation in Section 5. Later, Section 6 gives a brief discussion and plans for future work, followed by a conclusion in Section 7.

2. Background and related work

A Clinical Decision Support System (CDSS) is an interactive computer software designed to assist health professionals with decision making tasks, such as preventing adverse drug events at the point of care [51]. Although many methodologies are employed in designing CDSS, including Bayesian networks, neural networks, and genetic algorithms, we focus on rule-based approaches in this study. Rule-based CDSS, derived from expert systems research [8], are knowledge based systems that integrate a medical knowledge base, patient data, and an inference engine to generate case specific advice. An overview of the state of the art of rule-based CDSS is given in [8,42,49].

Although CDSS are said to be very helpful to assist in the clinicians' work, still they are reluctant to adopt CDSS because of unnecessary workflow disruptions [26]. The importance of integrating a CDSS with a clinical workflow has been stressed by other studies as well [17,26,52]. Kawamoto et al. [27] note that CDSS interventions that are presented automatically and fit into a workflow of medical staff are more likely to be adopted, and a CDSS that makes recommendations is better than one that only gives an assessment. Another challenge is to ensure the correctness of decision support by following standard guidelines. The objective of a CDSS is to deliver the "*right knowledge* to the *right people* in the right form *at the right time*" [47]. Thus, integrating medical guidelines along with clinical data into a CDSS aims to deliver evidence-based recommendations to care providers at various points of care. Table 1 summarizes a variety of approaches from different research communities and compares them in terms of their knowledge base organization and workflow integration capability. Further details of these studies are discussed next along with the role of ontology in healthcare.

2.1. Clinical workflow

The business process management community has devoted a lot of effort in designing customized and flexible clinical processes using formal workflow languages (e.g., BPMN and BPEL) that are executable by existing workflow engines. Lenz and Reichert [30] conducted a survey of IT support for healthcare processes and emphasized the importance of flexible workflow design in supporting clinical decisions. Such systems have been developed to enable dynamic changes in predefined process models, such as ADEPT$_{flex}$ [46] and AgentWork [33]. Research on context-aware workflow design [2,6,20] also belongs to this area, but focuses more on the integration of context in binding specific services or constructing subprocesses, whereas medical knowledge is deemphasized. These techniques are useful and highlight the criticality of context in designing a flexible workflow.

However, there has also been work from the software engineering community on modeling medical processes. In particular, limitations of current workflow languages have been observed [56], and an alternative approach based on Little-JIL language [53] has been proposed. Little-JIL is a language that centers on a coordination diagram of the process described by a hierarchical task decomposition. It helps to coordinate agents and their activities, and allows steps to be performed in sequence or parallel. A Little-JIL description can be verified using finite state machine verification techniques. Approaches for formally defining and analyzing medical processes using Little-JIL are discussed in [11,12,39]. This stream of research is useful for its focus on improving patient safety and detecting medical errors, but it does not emphasize flexibility or decision support. Mathe et al. [32] developed the Model-Integrated Clinical Information System using model-based design techniques to represent complex clinical workflows in a service-oriented architecture. Thus, treatment protocols are transformed into executable constructs such as Web services or BPEL processes to promote software reuse and maintainability.

2.2. Clinical guidelines

The medical informatics community models clinical processes as guidelines or care plans. Their focus is on medical decision making by interpreting situations based on best practice. They also formalize process definitions in a way that reflects clinical tasks and constraints as clinicians perceive them.

Medical guidelines were originally expressed as free-format text documents to assist medical decision-making during diagnosis, management and treatment within different areas of healthcare. Recently, various approaches have been proposed to represent Computer-Interpretable Guidelines (CIGs), such as Arden Syntax, Asbru, EON, GLIF, GUIDE, PRESTIGE, PRODIGY, PRO*forma*, and SAGE (see [17, 22, 36, 38, 43, 44]). CIGs can produce personalized recommendations during patient encounters and reduce variance in patient treatment. Peleg et al. [44] reviewed six CIG modeling approaches and established a consensus on their common structure. They represent clinical guidelines as *plans*, whose components represent decisions, actions, and the relationships among them. *Decision steps* are used for conditional and unconditional routing of the flow, while *action steps* are used to specify a set of tasks or a sub-plan to be carried out. A review of systems using CIGs was conducted in [24]. The main disadvantage is that all CIGs require an execution engine that is not freely available. Complex workflow patterns cannot be easily modeled in this way, and validation (e.g., checking for possible deadlocks) of such workflows is difficult.

2.3. Ontology in healthcare

An ontology is a formal specification of the concepts within a domain and their interrelationships. The Web Ontology Language

Table 1
Comparison of existing approaches for modeling clinical processes.

Approach	Criteria	Knowledge/data source			Workflow integration	
		Form of clinical data	Clinical guidelines	Formal vocabulary	Workflow flexibility	Workflow patterns
Process modeling approach	Workflow modeling [2,6,20,33,46]	Unstructured	---	---	++	++
	Ontology modeling [3,10,13]	Ontology	+	---	++	+
Software engineering approach	Little-JIL [11,12,39]	EMR	++	---	++	+
	Model integrated approach [31,32]	EMR	++	---	+	+
Guideline modeling approach	CIGs [17,22,38,43,44]	N/A	++	N/A	N/A	N/A
	Ontology-based approach [1,55]	Unstructured, EMR	++	---	++	+

Notation: N/A: not applied; --- not supported/considered; +: weakly supported; ++: strongly supported.