Decision Support Systems 54 (2012) 781–794

Contents lists available at SciVerse ScienceDirect

Decision Support Systems

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

A framework for transformation from conceptual to logical workflow models

Shaokun Fan [a,*], J. Leon Zhao [a], Wanchun Dou [b], Manlu Liu [c]

[a] Department of Information Systems, City University of Hong Kong, Kowloon, Hong Kong, China
[b] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[c] E. Philip Saunders College of Business, Rochester Institute of Technology, Rochester, NY, United States

ARTICLE INFO

Article history:
Received 19 March 2011
Received in revised form 18 July 2012
Accepted 5 September 2012
Available online 14 September 2012

Keywords:
Business process
Workflow pattern
Conceptual model
Logical model
Transformation
Validation

ABSTRACT

Both conceptual and logical workflow models are needed to support business process automation via workflow systems. Conceptual models are normally used to document the generic business process requirements in the company. Logical models are generally used for defining technology specific requirements, where software modules as well as their behavioral patterns should be clearly specified. However, the transformation from conceptual models to logical models can be a tedious task, often causing errors in the resulting logical model. In this paper, we propose a formal approach that can be used to support efficient and accurate model transformation. First, we develop a procedure for transforming a conceptual workflow model into its corresponding logical workflow model. Business requirement analysis, dependency mapping, and workflow pattern-based model transformation are the major components of this transformation procedure. Second, we create a validation procedure that can validate whether the derived logical model is consistent with its original conceptual model. Business process ontologies are employed in our approach to describe both conceptual and logical models. We also implement a prototype system and conduct a demonstrative case study to show the feasibility of our approach.

1. Introduction

Business process modeling and workflow technologies have become essential when developing enterprise information systems. There are various process modeling languages to describe processes [34]. While much attention has been paid to the logical correctness of these models [4,41], developing a workflow application that can fulfill given business requirements is also very important [16,38]. The terms "business process model" and "workflow model" are both found in the literature. The business process model is often used when communicating with managers, while the workflow model is commonly used at the system level. In this paper, we will use both terms interchangeably.

This research is motivated by the need to resolve a real-world problem in the context of the Kuali project [21]. Kuali is a community source project to develop a comprehensive suite of administrative software that meets the needs of all Carnegie Class institutions. There are currently more than twenty development partners in the Kuali project. In this context, we need to develop a workflow model to support software change management based on a conceptual business process model. Further, we need to validate that the workflow model we develop is consistent with the given conceptual business process model. However, we could not find an existing approach for systematically transforming a conceptual business process model to a physical workflow model.

Over the past twenty years, a lot of work has been done in the area of business process modeling. Much research has been devoted to model expressiveness [32,42], and some research has focused on business process model verification [4,34,41,46]. However, these approaches stop at logical correctness. Only a few approaches [16,23,38] in the literature explicitly capture business requirements in the workflow design process even though doing so was suggested ten years ago [10]. Further, for formal verification, some workflow models are very difficult for managers to understand, which often results in a gap between managerial users and technical developers of workflow applications. For example, in order to add a new task to a Petri net-based workflow model, one must manipulate the model in terms of transitions, places, arcs and tokens, which can be done correctly and efficiently only by someone well-versed in Petri nets, a skill not normally possessed by ordinary managers.

Designing a workflow model is a knowledge-intensive endeavor because creating a typical workflow model requires detailed understanding of various process components, such as business process logic, the organizational chart, and the information systems accessed by the workflow. The whole design process may require collaboration between an enterprise's functional and technical departments. More importantly, the model is subject to frequent modification due to changes in the process components. As has been done in the database field, dividing the design process into three phases, namely conceptual, logical, and physical design, should enhance the efficiency of modeling

* Corresponding author at: P7904, Academic Building, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, China. Tel.: +852 34427540; fax: +852 34420370.
E-mail address: Shaokfan@cityu.edu.hk (S. Fan).

0167-9236/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2012.09.006

as well as the quality of the design output. A conceptual, logical, or physical business process model is the output of each design phase respectively. The conceptual business process model has a higher level of abstraction than the other two types of models. The transformation from conceptual business process model to logical business process model and then to physical business process model is very important in terms of mapping business requirements to system implementation. The terms of "conceptual model", "logical model" and "physical model" are used to represent the three models in the rest of this paper.

In this paper, we present a detailed transformation procedure from conceptual to logical models. We choose Dependency Network Diagrams (DND) [37] as the conceptual model because of its simplicity and expressiveness. Further, Petri nets are chosen as the sample logical modeling language because of the availability of abundant verification techniques [40]. Here, the conceptual model is mainly used to capture the business requirements in enterprises; the logical model is used for information system (e.g. workflow) design purposes; and the physical model is only used for system execution. In particular, the key challenges in this research are how to derive a logical model from a given conceptual model and how to validate that the derived logical model is consistent with the given conceptual model. To provide a flexible modeling framework, model designers can derive logical models iteratively based on conceptual models in the logical design process. That is, logical design of a workflow model can involve multiple logical models.

The main contributions of this paper are three-fold. First, we propose a three-layer modeling approach that differentiates among conceptual, logical and physical models. Second, we develop a methodology for transforming a conceptual model into a logical model. Third, we create an approach for validating whether the derived logical model is consistent with its corresponding conceptual model. The rest of this paper is organized as follows. In Section 2, we review the related areas of this research. In Section 3, we define conceptual, logical and physical models in detail and we address relationships among the three types of models. Section 4 gives an example conceptual model in *DND*. Section 5 presents the transformation procedure for deriving a logical model from a conceptual model and introduces the validation procedure. Section 6 validates our approach by a case study and prototype system development. We conclude in Section 7 with a discussion of the results and limitations of our research.

2. Literature review

2.1. Business process modeling

Business process modeling has been a subject of study from both managerial and technical perspectives. From the managerial perspective, business process modeling is about the understanding and analysis of business processes. Over the past twenty years, business process modeling became an important aspect of Business Process Redesign (BPR) for business management in order to improve business efficiency [1,29]. The focus of business process modeling in BPR is on "Why" a particular process activity is undertaken [5]. For instance, Hammer identified seven principles that should guide any business process re-engineering exercises undertaken [12]. Typically, the result of business process modeling is a model at the *conceptual level* since no consideration is given to what technology to use in the implementation of the given business processes. Popular modeling languages from this area include GED [18], i* model [48], and DND [37].

From the technical perspective, business process models provide a blueprint for the development of information systems, leading to model-driven system development [2]. A business process model is also referred to as a workflow model although a workflow model typically requires detailed information in five perspectives, namely, functional, behavioral, informational, operational, and organizational [34]. This is because a workflow model needs to be deployed and executed in workflow management systems (WFMS) while a business process model might not [49,50,52]. Consequently, a workflow model requires specific information related to the workflow technology (*logical level*) or even specific workflow software (*physical level*). Important modeling languages from this area include Petri nets, UML activity diagrams, and BPMN.

Little work has been done to explore the relationship between the managerial and technical perspectives [8]. Existing process modeling languages that feature different degrees of abstraction for different user groups exist and are used for different purposes in business process management [8]. Through a case study on process modeling, Glassey identified three levels of process models: the abstract level, the organizational level, and the operational level [11]. Similarly, Dreiling et al. [8] distinguished three perspectives in process modeling: management, business process analyst, and technical analyst. Our work is similar to these studies because we emphasize the difference between different types of models. However, our approach also tries to formalize three levels of workflow models and facilitate model transformation.

A couple of methods have been proposed recently to develop business process models based on particular business requirement documents. In [16,23,38], van der Aalst et al. use Colored Petri nets (CPN) as a requirement model to specify, validate, and elicit user requirements. Then the requirement in CPN is transformed to a workflow model and to an implementation of the new system. These approaches hold similar objectives, but they use the same model language (i.e. CPN) to describe both user requirements and workflow models. The drawback of this method is that managers do not understand CPN. Our approach chooses conceptual models in the business process analysis domain as the starting point and helps business process modelers design business models to meet the business requirements.

In this paper, DND and Petri nets are chosen as examples of conceptual and logical modeling languages, respectively. DND [37] was recently proposed as a new representation methodology, which allows the essential elements governing organizational relations to be captured, communicated, and evaluated under changing conditions. By depicting important features of organizational relations, information systems can be designed explicitly for the control and coordination of organizational activities. Petri nets, as a state based graphical modeling language, have become one of the most popular workflow modeling languages [40]. Many analysis techniques are available for Petri nets. Thus, DND and Petri nets are chosen as the example modeling languages in this paper.

2.2. Business process model transformation

The transformation between models of different levels of abstraction such as platform-independent models and platform-specific models is a critical step of system development in model-driven architecture [2]. While model transformation techniques have attracted lots of attention [6], defining a transformation between any two workflow modeling languages is still a difficult task as several domain-specific problems remain to be solved. In [26], seven issues about defining business process model transformations are identified based on the observations of four business process modeling languages.

Some approaches have been proposed for transforming one workflow modeling language to another. In order to perform formal analysis on BPEL, both BPEL2PN [14] and WofBPEL [28] provide the functionality of transforming BPEL to Petri nets. BPMN, as a popular workflow modeling language, can be translated to Petri nets through certain mapping rules [7]. BPMN can also be translated to BPEL for the purpose of system implementation [27]. Other transformation approaches for workflow modeling languages can also be found in the literature. However, these approaches are mostly done in an ad-hoc way.

In addition to the language-specific approaches mentioned above, there are a number of other approaches to develop a general framework for business process model transformation. Lohmann et al.