Information Systems 42 (2014) 195–215

Contents lists available at ScienceDirect

Information Systems

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

How to guarantee compliance between workflows and product lifecycles?

Zhaoxia Wang [a,b,c,d], Arthur H.M. ter Hofstede [e,f], Chun Ouyang [e], Moe Wynn [e], Jianmin Wang [a,b,c,*], Xiaochen Zhu [a,b,c]

[a] School of Software, Tsinghua University, Beijing, China
[b] Key Lab for Information System Security, Ministry of Education, Beijing, China
[c] National Laboratory for Information Science and Technology, Beijing, China
[d] Logistical Engineering University, Chongqing, China
[e] Queensland University of Technology, Brisbane, Australia
[f] Eindhoven University of Technology, Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 3 June 2011
Received in revised form
14 November 2013
Accepted 15 January 2014
Recommended by M. Weske
Available online 23 January 2014

Keywords:
Product lifecycle management
Workflow management
Verification
Process model
Compliance

ABSTRACT

Product lifecycle management (PLM) systems are widely used in the manufacturing industry. A core feature of such systems is to provide support for versioning of product data. As workflow functionality is increasingly used in PLM systems, the possibility emerges that the versioning transitions for product objects as encapsulated in process models do not comply with the valid version control policies mandated in the objects' actual lifecycles. In this paper we propose a solution to tackle the (non-)compliance issues between processes and object version control policies. We formally define the notion of compliance between these two artifacts in product lifecycle management and then develop a compliance checking method which employs a well-established workflow analysis technique. This forms the basis of a tool which offers automated support to the proposed approach. By applying the approach to a collection of real-life specifications in a main PLM system, we demonstrate the practical applicability of our solution to the field.

1. Introduction

Product lifecycle management (PLM) systems [32] play an important role in facilitating collaboration and in improving product development in the manufacturing industry. The main functionality offered by such systems centres around product data management (PDM) [30] and process management. Version control of product objects (e.g. blueprint, design specification) is at the core of PDM. Given a predefined set of object version operations (e.g. check in, check out, release), the application of these version operations may change the state of the objects to which they are applied, and the state changes are often subject to constraints enforced by the version control mechanism. In PLM, there are also business processes (e.g. product design) that involve tasks which need to access and/or update the related objects. Compliance between processes and object version control policies can be addressed by maintaining the state of objects and restricting the application of version operations on objects based on their states as well as by maintaining access privileges for the various tasks in the processes. For example, an object that has been checked out at a certain task cannot be modified by any other task till it has been checked in again.

Contemporary PLM systems typically use workflow technology to provide support for the management of processes. Many common business processes in the manufacturing

* Corresponding author at: School of Software, Tsinghua University, Beijing, China. Tel.: +86 10 62781776.
E-mail addresses: wang-cx06@mails.tsinghua.edu.cn (Z. Wang), a.terhofstede@qut.edu.au (A.H.M. ter Hofstede), c.ouyang@qut.edu.au (C. Ouyang), m.wynn@qut.edu.au (M. Wynn), jimwang@tsinghua.edu.cn (J. Wang), zhu-xc10@mails.tsinghua.edu.cn (X. Zhu).

0306-4379/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2014.01.006

industry (e.g. in areas such as engineering design, product release, and production control) involve the use of object version operations. The use of these operations in the context of tasks is subject to access control restrictions. In addition, the order in which these operations may be performed is often governed by an object versioning lifecycle which explicitly specifies that certain operations can only be applied when the object is in a certain state. As ordering relations between version operations are also implicitly enforced by the ordering relations between tasks in the process model, compliance issues may rear their head. More specifically, on the process side, one can specify access privileges for tasks in terms of which version operations are permitted during the execution of these tasks and how they may progress the state of objects, while on the object side, the versioning lifecycle prescribes which version operations can be performed in which state of the object. As such, it is possible that the ordering of version operations as implied by the order of tasks in the process may violate what is prescribed in the object versioning lifecycle.

Hence, a research question arises: *how can we determine compliance between processes* (*or workflows*) *and the relevant object versioning lifecycles in production lifecycle management*? Addressing this question is not trivial and the challenges lie in two main facts. Firstly, task ordering relations in processes can be complex. Secondly and more importantly, processes and object versioning lifecycles are defined in two different areas (i.e. process management and product data management). On one hand, object versioning lifecycles and the ordering of tasks in processes are specified independent of each other. On the other hand, tasks in processes often need to perform certain version operations in object versioning lifecycles subject to access control rules. So far, in the domain of product lifecycle management, access control in data management and in process management is disconnected and consistency between access control rules in these two parts is not guaranteed yet.

In the field of business process management, there are existing studies on checking the consistency between business process models and object lifecycles [34,35]. Their approach is based on the fact that business process models and object lifecycles represent two different views of the *same* system. The tasks in process models and the state transitions in object lifecycles are defined over a common set of actions. However, in the research question we face, process models and object versioning lifecycles capture behaviours of *different* systems. The tasks in processes and the state transitions in object versioning lifecycles are defined in two separate areas: the former represent actions in process management, while the latter capture version operations in product data management. The problem thus cannot be solved using existing approaches as reported in [34,35].

This paper aims to address the issues involved in determining compliance between product data management and process management in PLM systems. It provides a solution to tackle the problem in a systematic and rigorous manner by formally defining the *notion of compliance* between process models and object versioning lifecycles and then develops a *compliance checking method*. This method employs a well-established analysis technique from the field of workflow management. A tool is then developed offering *automated support* to the proposed approach. Finally, the approach is applied to a number of *real-life specifications* in a main PLM system providing insight into its potential practical applicability.

Our solution, as strongly driven by the requirements of the domain of product lifecycle management, is straightforward in tackling the non-compliance problem between process management and product data management and can be directly applied to improving the design of access control rules in this domain. Our findings also contribute to the research topic on consistency checking between business processes and object lifecycles in the field of business process management by expanding the problem to include the situation where these two artifacts correspond to independent systems. As our approach builds on a generic formal analysis framework, we believe that it is possible to further extend it to deal with compliance checking between processes and object lifecycles in general rather than just in the context of PLM systems.

Our research has been carried out in line with the design science methodology [19]. In the following outline of the paper, the specific guidelines of the methodology that apply to various sections in the paper are included in brackets. Firstly, Section 2 introduces the background knowledge (*Problem Relevance*) and provides a formal definition of compliance between a versioning-aware process model and an object versioning lifecycle (*Design as an Artefact*, *Design Rigor*). Based on that, Section 3 proposes an approach for automatic compliance checking (*Design as an Artefact*), and proves the correctness of this approach (*Design Rigor*). Subsequently, Section 4 presents the development of tool support (*Design Evaluation*) and Section 5 discusses its application to a collection of real-life specifications (*Design Evaluation*). Section 6 provides a review of related areas and comparison to relevant research efforts (*Design Contributions*, *Search Process*). Finally, Section 7 summarises our current research findings (*Design Contributions*) and outlines the future work (*Search Process*).

2. Fundamentals

In this section background information on version management and workflow management is provided in order to be able to precisely characterise the problem and its proposed solution.

2.1. Version management

Version management (or version control) [50] is widely used in the management of engineering data [49]. Version management is concerned with maintaining different versions of objects and configurations and with providing support for operations on these versions. The scope of object version management is a single object, e.g. a specific car design, while configuration version management deals with the ways component designs can be combined to create more complex design artifacts. As many business processes supported by PLM systems are concerned with