Computers in Industry 67 (2015) 72–85

Contents lists available at ScienceDirect

Computers in Industry

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

Mining version histories for change impact analysis in business process model repositories

Hoa Khanh Dam [*], Aditya Ghose

School of Computer Science and Software Engineering University of Wollongong, Australia

ARTICLE INFO

Article history:
Received 3 September 2013
Received in revised form 15 August 2014
Accepted 21 October 2014
Available online 3 December 2014

Keywords:
Business process change management
Change impact analysis
Mining version history

ABSTRACT

In order to remain competitive and sustainable in today's ever-changing business environments, organizations need to frequently make changes to their business activities and the corresponding business process models. One of the critical issues that an organization faces is change impact analysis: estimating the potential effects of changing a business process to other processes in the organization's business process repository. In this paper, we propose an approach to change impact analysis which mines a version history of a business process model repository. Our approach then identifies business process models that have been co-changed in the past and uses this knowledge to predict the impact of future changes. An empirical validation on a real business process model repository has showed the effectiveness of our approach in predicting impact of a change.

1. Introduction

A business process is defined as consisting of a set of activities, performed by their relevant roles or collaborators, to intentionally achieve a set of common business goals [1]. Business processes are the core assets of any enterprise, covering many aspects in industry such as design, engineering, manufacturing, purchasing, physical distribution, production management and supply chain management. Organizations committed to long-term business process management (BPM) may have repositories of hundreds or even thousands of business process models. For example, the IBM BIT Process Library has 735 process models [2], the SAP Reference Model contains 604 process models [3], and there are 6000+ process models in Suncorp's process model repository for insurance [4]. On the other hand, the ever-changing business environment (due to various reasons such as new customer requirements, global competition pressures, new regulations, new IT solutions, economic down turn, etc.) demands organizations to constantly consider changing their business activities in order to remain competitive and sustainable in the long term.

Business process models are essential knowledge assets for an organization (with hundreds and thousands of business process models [5]) to manage its business processes in terms of documenting and implementing procedures, control their execution, analyze their performance, and improve them (i.e. business process management [6,7]). Recent studies (e.g. [8]) have demonstrated various perceived benefits of using business process models as the basis for process improvement, understanding, communication, execution, analysis and simulation. In particular, both the practitioner and vendor groups participating the study conducted in [8] agreed that support for continuous improvement of an organization's business processes (in order to react to changes in its business environment) is the core benefit of business process models. Changes in process models need to be put into practice immediately in order for process improvement to be effective [1]. Recent work (e.g. [9]) provide techniques for automatic execution of business process models or develop process engines that can interpret process models and enact them automatically. The explicit documentation of business processes in those contexts facilitates change management in terms of quickly identifying what needs to be changed and implementing those changes rapidly.

However, making changes to large, complex repositories of business process models is a highly challenging task. This is mainly due to the *ripple effect* caused by a change. Specifically, a change made to one business process can potentially affect a range of other processes that are related to the process being changed. For example, changes initially made to a sub-process (e.g., adding a new activity) may lead to secondary, additional changes made to the processes that contain this sub-process. Such changes made to those processes may lead to further changes in other related

[*] Corresponding author. Tel.: +61 0242214875.
E-mail addresses: hoa@uow.edu.au (H.K. Dam), aditya@uow.edu.au (A. Ghose).
URL: http://www.uow.edu.au/_hoa/, http://www.uow.edu.au/_aditya/

http://dx.doi.org/10.1016/j.compind.2014.10.005
0166-3615/© 2014 Elsevier B.V. All rights reserved.

processes. In a large repository of hundreds or even thousands of business process models, it becomes critical to determine the impact of a change, which is the core focus of our paper.

Change impact analysis usually starts with the business analyst examining the change request and determining the processes initially affected by the change (i.e. the *primary changes*). The business analyst then determines other process models in the business process repository that are potentially affected and required to be changed. Changes made to those impacted processes may also potentially affect other processes and thus the impact analysis continues this procedure until a complete impact set is obtained. Change impact analysis plays a major part in planning and establishing the feasibility of a change in terms of predicting the cost and complexity of the change (before implementing it). This helps reduce the risks associated with making changes that have unintended, expensive, or even disastrous effects on existing business operations.

Organizations face a range of challenges in managing change in the context of large and complex collections of business processes. While an important class of change management problems pertain to process instances, our focus in this paper is however exclusively on process designs/models. Here, we take a model-based approach to impact analysis which examines impacts to the business processes before the implementation of such changes. An appropriate decision can therefore be made (before any detailed implementation of the change is considered) on whether to implement a specific set of changes based on what business process models are likely to be impacted and thus on the likely change cost. Earlier decision making and change planning are clearly important in the context of rigorous change management. We also acknowledge that there may be a gap between the actual execution of a process and what is being described in its model, and change impact analysis for process instances is an alternative.

Although business process management research is gaining increasing attention from both industry and academia, there has been very little work on supporting change impact analysis in business process model repositories [5]. Some recent work (e.g. [10]) only focus on identifying dependency relationships among different entities in a single process in order to analyze the impact of a change made to one part of the process to other parts of the process. The work in [11] specifically aims to support change impact analysis between services and business processes in a service-oriented environment. The recent work in [12] addresses the issue of propagating changes to maintain consistency within a process repository, which is part of change implementation rather than change impact analysis.

Traditional approaches to change impact analysis in business processes tend to focus on establishing inter-process relationships (i.e. dependencies) and using this knowledge for impact analysis. These approaches rely on a classification of relationships between business processes (e.g., [13,12]). However, basic dependency-based impact analysis techniques are considered to be conservative in that they consider all possible transitive closure of inter-process relationships. Results produced by those techniques may have enormous impact sets, which are sometimes unnecessary or even too large to be of practical use [14]. In addition, establishing inter-process relationships in such a way that precisely reflect the semantic nature of dependency between processes (e.g., annotating process models with semantic effects as done in [15]) may be labour-intensive and time consuming.

In this paper, we propose an alternative approach which focuses on detecting factual inter-process dependencies as manifested in the evolution of the process models. Our approach mines the revision history of a process model repository, and identifies processes that have been frequently changed at the same time to identify *co-variation patterns* between them. This approach computes the impact based on the heuristic that *processes that have been changed together in the past (co-variation coupled) will be likely changed together in future*. We have performed an empirical validation using a real business process repository to compare the effectiveness of our approach against the basic dependency-based impact analysis technique.

The structure of this paper is as follows. In the next section, we briefly describe how business processes are defined using the standard Business Process Modeling Notation and how they are annotated with semantic effects. In Section 3, we present a generic framework for change impact analysis in business process repositories. We then discuss in detail a basic inter-process relationship (dependency-based) analysis approach (Section 4) and our revision history mining approach (Section 5). We report on our evaluation in Section 6 and discuss related work in Section 7. We then conclude and outline our future work in Section 8.

2. Background

2.1. Business Process Modeling Notation

While there are a range of modeling notations for business processes, for our purposes we use the Business Process Modeling Notation (BPMN) which is a standard for business process modelling [16]. It provides graphical notation for specifying various types of activities, decision responsibilities, control and data flow in business process within one organization and in cross-organizational settings. BPMN has been widely used in the industry due to its powerful notation which is readily understandable by both the business stakeholders and the technical developers. Fig. 1 shows a BPMN diagram describing a typical process of assessing insurance claims and a process of making claim payments.

There are three major categories of notational elements in BPMN: Flow Objects, Connecting Objects and Swimlanes. The elements known as Flow Objects are: Events, Activities, and Gateways. Activity is the most common type of Flow Objects which describes the kind of work which must be performed. A Task is a atomic activity while a Sub-Process is a composite activity, which contains additional levels of business process detail. In Fig. 1, "Investigate Claim" and "Reject Claim" are examples of activity tasks. A Gateway is used to control the divergence and convergence of Sequence Flow and determines branching, forking, merging, and joining of paths. There are different types of gateways (exclusive, inclusive, complex, and parallel) and the behavior of each type gateway specifies how many of the gates will be available for the continuation of flow. Fig. 1 has an XOR gateway modelling decisions that are based on the validity of a claim.

Flow Objects are connected in three different ways: by sequence flows, message flows or association. Sequence Flows are used to show the order in which a series of flow objects have to be completed. A Message Flow describes the exchange of messages between two process participants while an Association is used to associate information (e.g., an Artifact or text) with Flow Objects. Process participants are represented in a BPMN diagram as a Pool. For instance, there are two pools, i.e. "Claim Dept" and "Finance Dept", in the process in Fig. 1 and they exchange messages with one another, e.g., the "Determine Payout" task in the "Claim Dept" pool sends a message to the "Finance Dept" pool. In order to separate different processes, a pool can be sub-partitioned into swimlanes. Finally, Artifacts are used to provide additional information about the process. A typical artifact in BPMN is Data Object which represents data required or produced in an activity.