Towards Rich Change Management for Business Process Models

Christian Gerth
University of Paderborn
Zukunftsmeile 1
33102 Paderborn, Germany
gerth@uni-paderborn.de

Markus Luckey
University of Paderborn
Zukunftsmeile 1
33102 Paderborn, Germany
luckey@uni-paderborn.de

ABSTRACT

Business process models play an important role in the development of large IT systems, since they are easy to understand by all project stakeholders. High-level process models may be created by domain experts, which are stepwise refined in later development phases until they become executable. To establish such model-driven development (MDD) approaches in praxis, a comprehensive tool support of the complete model life cycle is necessary including model change management in particular. In this position paper, we give an overview of our framework for change management of business process models. This framework allows to merge process models in different modeling languages and considers the execution semantics of process models during comparison. Based on these results, we derive further research challenges with the aim to obtain a *rich change management* solution for business process models.

Keywords

Model-driven development, business process models, change management, model merging, semantics

1. INTRODUCTION

In model-driven development approaches of large IT-projects, business process models (BPM) play an important role and are used in different development phases. Due to their general comprehensibility among project stakeholders, process models help to bridge the gap between business and IT requirements.

Similar to other artifacts in software development, process models underlie constant change and are developed in team environments. As a consequence, based on a common source version V of a process model, different versions are created (V_1, V_2). At some point in time, e.g. when a local version shall be committed to a repository, the individually refined process models need to be merged by means of *process model change management* in order to obtain an integrated version V_M.

Process model change management comprises different activities, such as model matching, difference detection, dependency and conflict analysis, as well as model merging by resolving differences between process model versions. In addition, the semantics of process models must be considered to enable the detection of composite differences and the detection of precise conflicts.

In previous work [6, 5, 3, 4, 2], we have developed a framework for process model change management that addresses these activities. Our framework is based on an intermediate representation and can be used to merge process models in different modeling languages, e.g. BPMN [8] or BPEL [9]. Parts of the framework contributed to the Compare & Merge framework of the IBM WebSphere Business Modeler V 7.0[1], which was released as an IBM product.

In this position paper, we derive research challenges that need to be addressed to obtain *rich change management* for business process models. Such solutions cover the different aspects of model merging comprehensively to be applicable in various commercial use case. Among these aspects are the support of *complex mappings* between elements in different models, the *integration* in existing business environments, and the *active guidance* of users through the merge process, e.g. by providing conflict resolution recommendations.

In Section 2, we briefly introduce our framework for process models change management. In Section 3, we discuss research challenges towards rich change management solutions for process models, and finally, in Section 4 we conclude.

2. FRAMEWORK FOR BPM CHANGE MANAGEMENT

An overview of our framework for business process model change management is shown in Figure 1. The framework consists of seven main components, all of which are briefly described in the following. References to more comprehensive publications can be found in the descriptions of the individual components.

Abstraction to the IR: To enable the use of our framework for process models in different modeling languages (such as BPMN [8], UML Activity Diagrams [7], and BPEL [9]), we first abstract process models in a concrete language to an intermediate representation (IR) [3]. The IR is based on workflow graphs, which we additionally decompose into fragments. Fragments enclose nested subgraphs with distinguished behaviors. Thereby, we make the implicit structure of IR process models explicit and are able to harmonize block-oriented and graph-oriented process modeling languages.

[1]http://www.ibm.com/software/integration/wbimodeler/advanced/

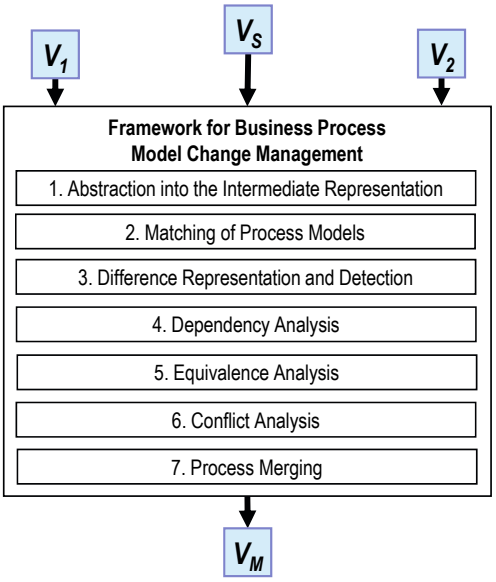

Figure 1: Framework for Process Model Change Management

Matching: In our matching approach, we match nodes and fragments of process models in the intermediate representation using matching strategies based on different similarity heuristics. Additionally, our approach is able to identify implicit relationships between splitting and joining gateways that constitute subgraphs with distinguished behaviors, such as parallel or alternative behavior.

Difference Detection: For the detection of differences between different process model versions, we apply an approach that results in a reconstructed change log consisting of compound change operations [6] that comprise several related elementary changes, e.g. the joint insertion of a fork and a join node that form a complete parallel fragment. In addition, compound change operation abstract from individual edge changes and take care that a process model remains connected after the application of a change operation. To enable an intuitive understanding of the differences, change logs are further organized hierarchically according to the implicit structure of the underlying process models.

Dependency Analysis: Compound change operations in a change log may depend on each other. That means, the application of a change operation requires the prior application of another change operations. These dependencies need to be computed before process models can be merged. For this purpose, we developed an approach that computes dependencies between compound change operations based on the process model hierarchy and on a dynamic specification of change operations [5]. Using this approach, different business process models can be integrated by applying change operations without being unnecessarily restricted to a certain application order.

Equivalence & Conflict Analysis: When process model versions are modified independently by applying change operations, these change operations might be in conflict. Two change operations are conflicting if the application of one operation turns the other one inapplicable. For conflict analysis, we first distinguished conflicting change operations into syntactic and semantic conflicts. Based on these conflict notions, we then introduced a method for conflict analysis between compound change operations [2].

Our method avoids the detection of false-positive conflicts between change operations that result in syntactically different fragments that are semantically equivalent. To that extent, we applied our approach that is based on a normalization of syntactically different but semantically equivalent fragments [4]. We decide equivalence by first transforming process models into process model terms. These terms are then normalized by a term rewriting system and finally the normal forms of the process model terms are compared syntactically. Our approach combines the benefits of syntactic and semantic comparison approaches to decide equivalences between process models and contained fragments. Our initial results have shown that taking the semantics of process modeling languages into account, helps to compute precise conflicts and avoids false-positive conflicts.

Merging: Finally, we merge different process model versions by applying compound change operations to resolve selected differences. To this extent, generic change operations on the IR are translated into language-specific change operations, which are applicable on process models in the underlying concrete modeling language.

We implemented prototypic tool support for most of the components in our framework. In addition, certain aspects of our framework are used in the *Compare & Merge Framework* in the commercial IBM WebSphere Business Modeler V.7.0[1]. Figure 2 shows a screenshot of the *Compare & Merge Framework*. In the top part of the user interface (see ②) a list of change operation representing the differences between a local and a remote process model version is shown. In the bottom part, the impact of selected changes is immediately visualized in the local version of the process model (see ③ in Figure 2).

3. RESEARCH CHALLENGES

In this section, we describe research challenges towards rich change management for business process models, starting with issues that are not (or only insufficiently) addressed by existing approaches.

Complex Mappings: The identification of complex mappings between groups of elements in business process models is only insufficiently addressed by existing matching approaches. A complex mappings occur e.g. when a single element is refined by a group of elements (1-n). Similar, an abstraction of multiple elements by a single element results in a complex mapping (n-1). The knowledge about refinements (or abstractions) clearly helps to grasp the actual intention of changes that have been applied during the evolution of models and hence help to understand differences between two models. A promising step towards the automated identification of complex mappings is described in the ICOP framework [10].

Integration in existing Business Environments: In enterprises and organizations a variety of information including guidelines, rules, and policies exist that constrain the development and evolution of business process models. For instance, business rules formalize constraints based on laws, standards, ethical, and cultural norms that might restrict the execution order of activities contained in business processes. If these business rules are formally specified, e.g. in terms of event-condition-action (ECA) rules, they could be validated automatically during the creation and merging of business process models. This approach would improve the quality of business process models, since the models would adhere immediately to the set of business rules.

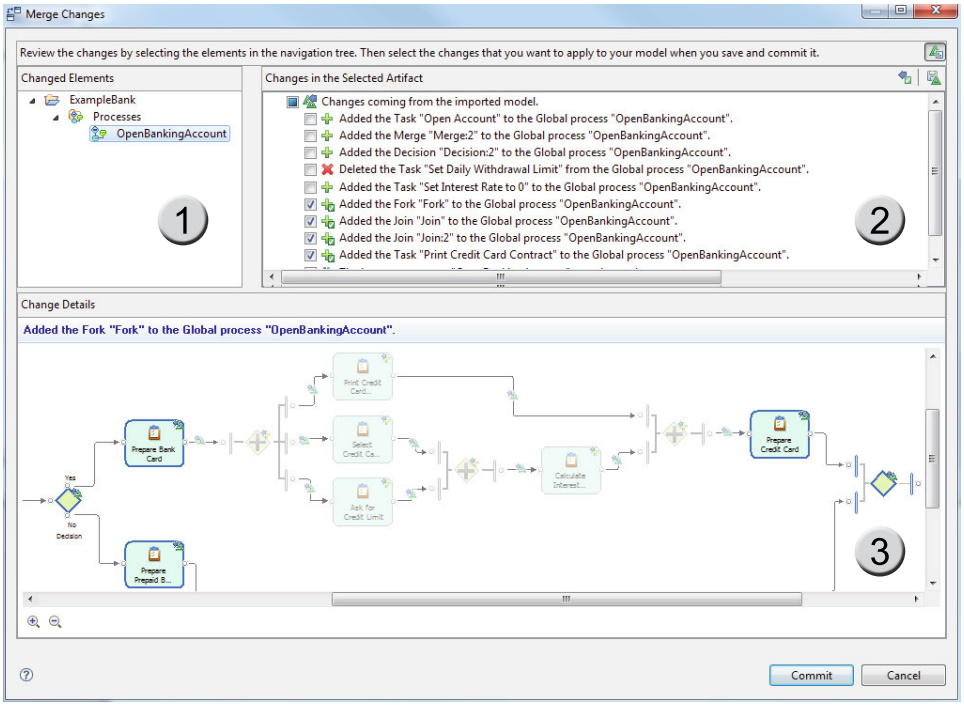

Figure 2: Compare and Merge Framework of the IBM WebSphere Business Modeler Version 7.0[1]

Active User Guidance: From the user perspective, change management of process models is complex, due to the multiple activities involved. As a consequence, users need to be actively guided through the process of change management. This is in particular important for activities that involve user interaction, such as the inspection of differences between business process models or the resolution of conflicts. For the former case, we already leverage the *"'What you see is what you get"'* principle (WYSIWYG) and directly preview the impact of selected changes in the merged business process model (see ③ in Figure 2). However, for conflict resolution active user guidance is currently missing. Here we intent to leverage the approach proposed by Brosch et al. [1] and provide recommendations for the resolution of conflicts based on predefined patterns.

4. CONCLUSION

The contributions achieved in our framework support the model-driven development (MDD) of complex software systems. We are convinced that MDD is a development methodology that is able to cope with the increasing complexity of today's software systems. However, MDD can only deliver its full potential if suitable tools are provided that support the distributed model development properly. To make these tools widely accepted, it is crucial to address also the different aspects of model merging comprehensively in terms of *rich change management*.

5. REFERENCES

[1] P. Brosch, G. Kappel, M. Seidl, K. Wieland, M. Wimmer, H. Kargl, and P. Langer. Adaptable Model Versioning in Action. In *Proceedings of Modellierung'10*, pages 221–236, LNI 161, 2010. GI.

[2] C. Gerth, J. Küster, M. Luckey, and G. Engels. Precise Detection of Conflicting Change Operations using Process Model Terms. In *Proceedings of MODELS'10*, volume 6395 (Part II) of *LNCS*, pages 93–107. Springer, 2010.

[3] C. Gerth, J. M. Küster, and G. Engels. Language-Independent Change Management of Process Models. In *Proceedings of MODELS'09*, volume 5795 of *LNCS*, pages 152–166. Springer, 2009.

[4] C. Gerth, M. Luckey, J. M. Küster, and G. Engels. Detection of Semantically Equivalent Fragments for Business Process Model Change Management. In *Proceedings of SCC'10*, pages 57–64. IEEE Computer Society, 2010.

[5] J. M. Küster, C. Gerth, and G. Engels. Dynamic Computation of Change Operations in Version Management of Business Process Models. In *Proceedings of ECMFA'10*, volume 6138 of *LNCS*. Springer, 2010.

[6] J. M. Küster, C. Gerth, A. Förster, and G. Engels. Detecting and Resolving Process Model Differences in the Absence of a Change Log. In *Proceedings of BPM'08*, volume 5240 of *LNCS*, pages 244–260. Springer, 2008.

[7] Object Management Group (OMG). Unified Modeling Language (UML): Activity Diagrams. http://www.omg.org/spec/UML/2.3, 2010.

[8] Object Management Group (OMG). Business Process Model and Notation (BPMN) Version 2.0. http://www.omg.org/spec/BPMN/2.0/, 2011.

[9] Organization for the Advancement of Structured Information Standards (OASIS). Web Services Business Process Execution Language (WS-BPEL) Version 2.0, 2007.

[10] M. Weidlich, R. M. Dijkman, and J. Mendling. The ICoP Framework: Identification of Correspondences between Process Models. In *Proceedings of CAiSE'10*, volume 6051 of *LNCS*, pages 483–498. Springer, 2010.