A Tool for Process Merging in Business-Driven Development

Jochen M. Küster[1], Christian Gerth[1,2], Alexander Förster[2], and Gregor Engels[2]

[1] IBM Zurich Research Laboratory, Säumerstr. 4
8803 Rüschlikon, Switzerland {jku,cge}@zurich.ibm.com
[2] Department of Computer Science, University of Paderborn, Germany
{gerth,alfo,engels}@upb.de

Abstract. Business-driven development favors the construction of process models at different abstraction levels and by different people. As a consequence, there is a demand for consolidating different versions of process models by merging them. In this paper, we study a basic scenario, derive requirements and present a prototype for detecting and resolving changes between process models.

1 Introduction

The field of business process modeling has a long standing tradition. Recently, new requirements and opportunities have been identified which allow the tighter coupling of business process models to its underlying IT implementation: In Business-Driven Development (BDD) [1], business process models are iteratively refined, from high-level business process models into models that can be directly implemented. As a consequence, business process models are a key artifact in BDD and advanced techniques for consolidating different versions of a process model are needed.

In general, such techniques for consolidating and merging process models have to provide means for identifying differences between versions of process models and resolving these by merging parts of process models. Specific techniques for process merging heavily depend on the underlying modeling environment. Existing work on process change management has focused mainly on the question of dynamic process changes where changes are made on already running processes [2, 3]. Solutions include techniques for migrating process instances to a new process schema and for identifying those cases where this is not possible. In these approaches, process changes are usually captured in a change log which is maintained by the process-aware information system [4]. Recent work by Weber et al. [5] introduces the concept of compound change operations (change patterns) for process models and compares existing workflow tools with regards to their support for process change management.

In contrast to existing work, our approach addresses the situation where no change log describing process model changes exists. This is a common situation in process modeling tools such as the IBM WebSphere Business Modeler [6] and also occurs in scenarios where process models are exchanged across tool boundaries.

In this paper, we first discuss a basic scenario for process merging and then derive important requirements for a solution. We then present key concepts of a prototype for process merging realized as a plug-in for IBM WebSphere Business Modeler.

2 Requirements for Process Merging and Tool Overview

Within business-driven development, process models are the central modeling artifacts. In this context, business process models are manipulated in a team environment and multiple versions of a shared process model need to be consolidated at some point in time. A basic scenario is obtained when a process model V_1 is copied and then changed into a process model V_2, possibly by another person. After completion, only some of the changes shall be applied to the original model V_1 to create a consolidated process model. Figure 1 shows an example process model V_1 that has been changed into a process model V_2.

Both models describe the handling of a claim request by an insurance company. V_1 starts with an *InitialNode* followed by the *actions "Check Claim"* and *"Record Claim"*. Then, in the *Decision*, it is decided whether the claim is covered by the insurance contract or not. In the case of a positive result the claim is settled. In the other case the claim is rejected and closed, represented by the *actions "Reject Claim"* and *"Close Claim"*.

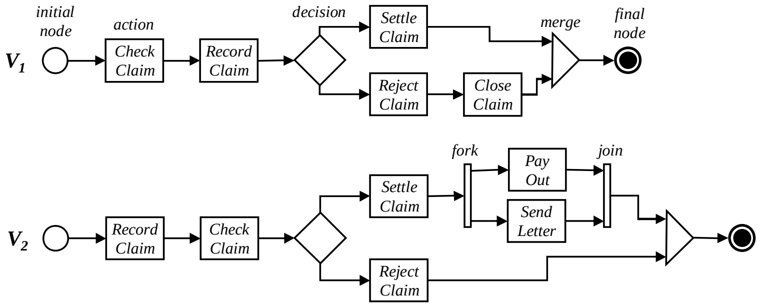

Fig. 1. Versions V_1 and V_2 of a business process model

Although process models V_1 and V_2 are similar at the first sight, there are some differences between the versions. The following differences can be detected:

- The positions of the *actions "Record Claim"* and *"Check Claim"* are changed.
- *Action "Close Claim"* does not exist in V_2.
- A new parallel structure (*Fork* and *Join*) is inserted in V_2 together with two *actions "Pay Out"* and *"Send Letter"*.

Process merging typically depends on the modeling language as well as on constraints of the modeling environment. In our case, the modeling language is given by the WebSphere Business Modeler which provides a language similar to UML 2.0 Activity Diagrams [7]. In our modeling environment, no syntax-directed editing of process models is performed and, as a consequence, also no change log is available. As such, in contrast to databases and existing approaches in process-aware information systems, there is no information about the performed changes on a process model. In the following, we describe the key requirements that a solution to process merging should fulfill:

- The solution must provide a technique to re-construct one possible change log which represents the transformation steps for transforming one process model into the other process model.
- The user should have the opportunity to select only some of the changes and apply it to the original model in order to obtain a new third model which can be considered as the merged process model.
- When applying changes, the user should not be restricted by prescribing a certain order whenever possible.
- Dependencies between change operations should be made explicit and taken into account when applying the changes. For example, when inserting a *Fork*, the corresponding *Join* should also be inserted in order to obtain a correct process model.
- The solution should provide user-friendly resolution of changes in the way that it reconnects inserted elements whenever possible and offers a possibility to perform related changes together at one time.

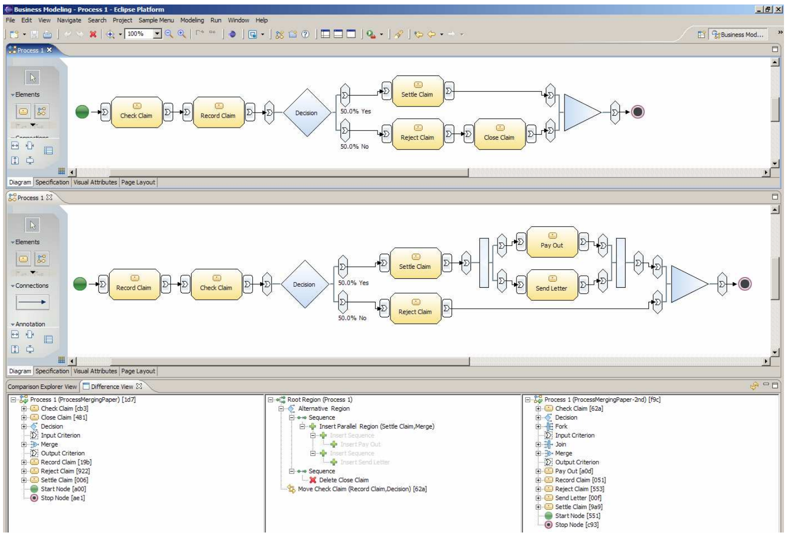

Fig. 2. Business Process Merging Prototype in the IBM WebSphere Business Modeler

Motivated by the requirements, we have developed an approach [8] for process merging which is divided into three steps. In the first step, we detect differences between the two process models using correspondences between model elements and the technique of Single-Entry-Single-Exit fragments (SESE fragments) [9]. In the second step, each detected difference is visualized according to the structure of the process models that is affected by it. The third step is then to resolve differences between the process models in an iterative way, based on the modeler's preferences. Here, for each difference, a resolution transformation is generated which resolves the difference between the two models and (if necessary) automatically reconnects the control flow.

The prototype has been implemented as an extension to the IBM WebSphere Business Modeler (see Fig. 2). It currently supports the following functionality [10]: copying of business process models, initial creation and update of correspondences, decomposition of process models into SESE fragments and detection of differences between two versions of a process model. In addition, the prototype provides several views that allow to visualize and resolve differences as well as to manipulate correspondences.

Fig. 2 shows versions V_1 and V_2 of the business process model introduced earlier in this paper. The lower third of Fig. 2 illustrates the Difference View, which is divided into three columns. The left and right hand columns show versions V_1 and V_2 of the process model in a tree view, which abstracts from control flow details of the process and focuses only on model elements of the process. The middle column of the difference view displays the differences between the two versions, which are arranged according to the structure of the process models and visualizes dependencies between differences. Using this view, the differences can be iteratively resolved with our prototype.

3 Conclusion

User-friendly process merging is a key technique for practical business-driven development. In this paper, we have first studied a basic scenario of process merging in BDD and established key requirements. We have presented our prototype, which visualizes differences between versions of process models and enables the resolution of differences, by applying change operations in an iterative way that automatically reconnect the control flow. Future work will include the elaboration of our approach for merging process models in a distributed environment. In those scenarios, the concept of conflict becomes important because one resolution can turn the other resolution non-applicable.

References

1. Mitra, T.: Business-driven development. IBM developerWorks article, http://www.ibm.com/developerworks/webservices/library/ws-bdd, IBM (2005)
2. Casati, F., Ceri, S., Pernici, B., Pozzi, G.: Workflow evolution. Data Knowl. Eng. **24**(3) (1998) 211–238
3. Rinderle, S., Reichert, M., Dadam, P.: Disjoint and Overlapping Process Changes: Challenges, Solutions, Applications. In Meersman, R., Tari, Z., eds.: CoopIS'04. Volume 3290 of LNCS., Springer (2004) 101–120
4. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Process-Aware Information Systems. Wiley (2005)
5. Weber, B., Rinderle, S., Reichert, M.: Change Patterns and Change Support Features in Process-Aware Information Systems. In Krogstie, J., Opdahl, A.L., Sindre, G., eds.: CAiSE'07. Volume 4495 of LNCS., Springer (2007) 574–588
6. : IBM WebSphere Business Modeler. http://www.ibm.com/software/integration/wbimodeler/
7. Object Management Group (OMG): UML 2.0 Superstructure Final Adopted Specification. OMG document pts/03-08-02. (August 2003)
8. Küster, J.M., Gerth, C., Förster, A., Engels, G.: Process Merging in Business-Driven Development. IBM Research Report RZ 3703, IBM Zurich Research Laboratory (2008)
9. Vanhatalo, J., Völzer, H., Leymann, F.: Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition. In: ICSOC 2007. Volume 4749 of LNCS., Springer (2007) 43–55
10. Gerth, C.: Business Process Merging - An Approach based on Single-Entry-Single-Exit Regions. Diplomarbeit, Universität Paderborn (October 2007)