Extending a Business Process Modeling Tool with Process Configuration Facilities: The Provop Demonstrator

Manfred Reichert[1], Steve Rechtenbach[1], Alena Hallerbach[2], and Thomas Bauer[2]

[1]Institute of Databases and Information Systems, University of Ulm, Germany
[2]Group Research and Advanced Engineering, Daimler AG, Ulm, Germany

Abstract. This tool demonstration presents an extension of the ARIS Business Architect in order to better cope with the high variability of business process models in practice. This extension is based on our Provop framework which provides sophisticated support for configuring and managing large collections of process variants. We have applied our tool in case studies in the healthcare as well as the automotive domain.

1 Motivation

One of the fundamental challenges for business process modeling is to cope with the multitude of variants that may exist for a particular process. In previous work, we introduced the Provop framework for configuring and managing such process variants at a high level of abstraction [1–3].

Provop provides an operational approach for variant configuration; i.e., a concrete process variant can be configured out of a predefined master process by applying a set of high-level change patterns [4] to it. Thereby, contextual information is utilized for enabling (semi-)automated variant configuration [5].

In the following we use the process of handling *vehicle repair in a garage* as running example (cf. Fig. 1a). In our tool demonstration we additionally provide examples from two case studies which we conducted in the automotive industry and the healthcare domain; i.e., we demonstrate how the process variants dealing with change management in automotive engineering and medical procedures in a hospital can be captured in our approach.

Our running example starts when receiving a vehicle. After a diagnosis is made, the vehicle is repaired if necessary. During diagnosis and repair the vehicle is maintained; e.g., oil and wiper fluid are checked. The process ends when handing over the vehicle back to the customer. Depending on the context (i.e., country-, garage- and vehicle-specific variables), different process variants are needed. Fig. 1b -Fig. 1d show three of them: Variant 1 (cf. Fig. 1b) assumes that the damaged vehicle requires a checklist of "Type 2" to perform the diagnosis. Thus, activities `Diagnosis` and `Repair` are adapted by modifying attribute `Checklist` to value "Type 2". Additionally, the garage omits maintenance of the vehicle as this is considered as special service not offered together with the

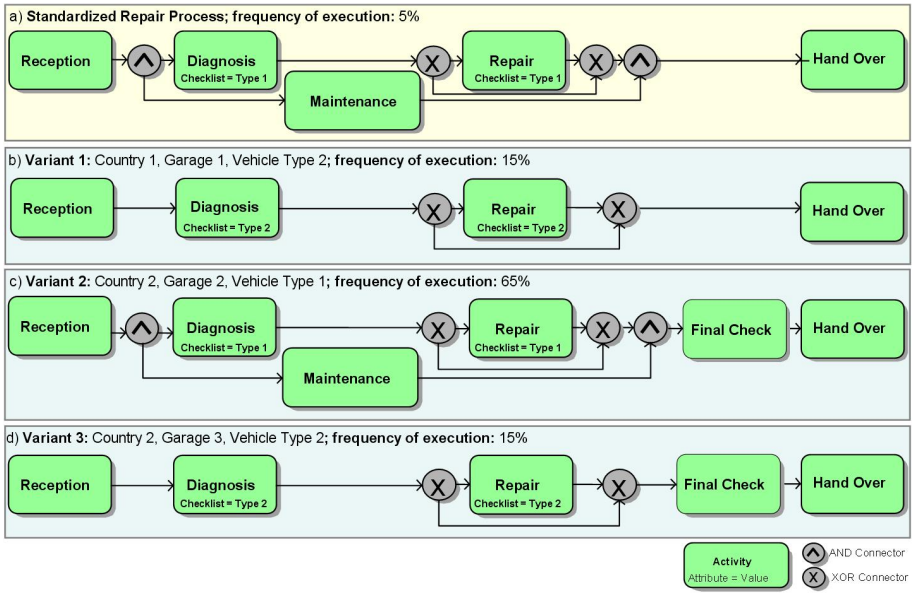

Fig. 1. Variants of a vehicle repair process in a garage (simplified view)

repair process. At model level this is realized by skipping activity `Maintenance`. Consider now Variant 2 (cf. Fig. 1c): due to country-specific regulations, a final security check is required before handing over the vehicle to the customer; i.e., activity `Final Check` has to be added when compared to the master process from Fig. 1a. Finally, Variant 3 (cf. Fig. 1d) will become relevant if a checklist of "Type 2" is required for diagnosis and repair, the garage does treat maintenance separately, and there are legal regulations requiring a final security check.

Note that in practice, hundreds of variants exist for the vehicle repair process. Contemporary BPM tools do not provide adequate support for modeling and maintaining such a large number of process variants. Generally, configuring process variants constitutes a non-trivial challenge when considering the many syntactical and semantical constraints the configured process variants have to obey in a given application context. One challenge is to design a master process that can serve as reference for configuring a family of related process models. Another one is to design, model, and structure the adjustments that may be applied to configure the different process variants out of this master process.

This tool demonstration picks up our Provop framework and shows how it can be applied to an existing BPM tool in order to support the modeling and management of process variants. Section 2 summarizes basic concepts of Provop and Section 3 gives insights into the Provop demonstrator.

2 The Provop Framework for Modeling Process Variants

Generally, a process model variant (*process variant* for short) can be created by cloning a given process model and adjusting it according to the specific requirements of its application context [3]. Provop has adopted this metaphor (cf. Fig. 2). A particular process variant can be configured by applying a set of predefined adaptations to a common master process (denoted as *base process* in Provop). For describing corresponding model adaptations, Provop supports well-defined change patterns: INSERT / DELETE / MOVE process fragment and MODIFY process element attribute.

In Provop a base process may be associated with adjustment points that correspond to the entries or exits of activities and connector nodes (e.g., split nodes) respectively (cf. Fig. 2). This enables designers of process adaptations to refer to specific model fragments. Using explicit adjustment points we can restrict the regions of the base process to which adaptations may be applied when configuring a variant. Finally, to enable more complex process adaptations as well as their reuse in different context, Provop allows to group change operations into reusable operation sets, which we denote as *options*: i.e., a particular variant is configured by applying a subset of the pre-modeled options to the base process. Thereby, Provop ensures soundness of the configured variant models [3].

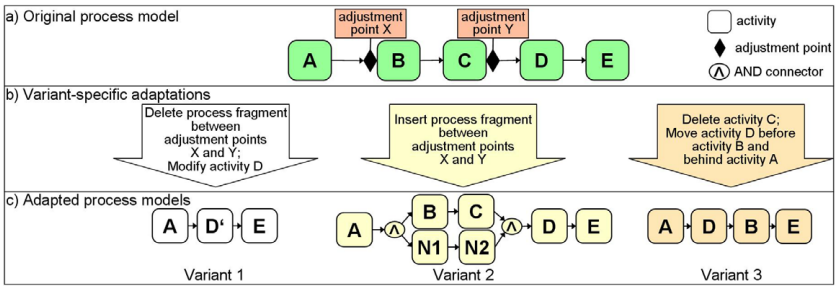

Fig. 2. General approach for variant modeling

Fig. 3 shows a base process with 3 corresponding options. This base process comprises 4 adjustment points to which options may refer; e.g., Option 3 refers to adjustment points `Repair Completed` and `Ready for Hand Over`, which then serve as anchors for the activity to be inserted by this option. When applying Options 1 and 2 together to the base process, for example, we obtain Variant 1 (cf. Fig. 1b). The model of Variant 2 (cf. Fig. 1c) may be created by applying Option 3 solely. Finally, Variant 3 (cf. Fig. 1d) results when first applying Option 2 and then Option 3 to the base process from Fig. 3a.

Provop provides additional features enabling the automated configuration of process variants. They utilize the process context and consider semantic constraints regarding possible adaptations of the given base process [1–3].

3 The Provop Demonstrator

Our Provop tool demonstrates how an existing BPM tool can be enriched with features for process variant management. Respective support is urgently needed in practice. Using the examples from our case studies and the described one, we demonstrate how to model a base process, how to define pre-defined adjustment of this process (i.e., options), and how to configure concrete variants by applying a subset of these options to the given base process. The latter includes soundness checks and context-based user assistance.

When realizing our Provop demonstrator we decided to use an existing BPM tool as implementation basis and to enhance it with facilities for process variant configuration and management. We have selected *ARIS Business Architect* for this purpose. *ARIS Business Architect* supports a variety of modeling notations (e.g., EPC and BPMN) and is widely used in practice for modeling, analyzing and optimizing business processes. The general limitations of contemporary BPM tools in respect to variant modeling also apply to the current *ARIS Business Architect* version, which enables the creation of new process variants by copying an existing model repository and by modifying its objects afterwards. However, this approach results in high redundancies of model data.

Another decision we made when implementing our Provop demonstrator concerns the choice of the meta model for representing base processes, corresponding options, and process variant models. We first applied Event Process Chains (EPCs). Unfortunately, EPCs do not offer grouping functions, which are highly relevant in our context in order to be able to group parameters of a particular change operation as well as to group change operations (into options). To enable grouping in *ARIS Business Architect*, in principle, model folders may be used as workaround. However, we decided to use the BPMN notation instead since it provides different grouping mechanism as required in our approach.

In our Provop demonstrator, basically, each option is realized as single BPMN model. Within these models corresponding operations of an option are encapsulated in "pools"; i.e., graphical and logical containers. Relevant parameters of a

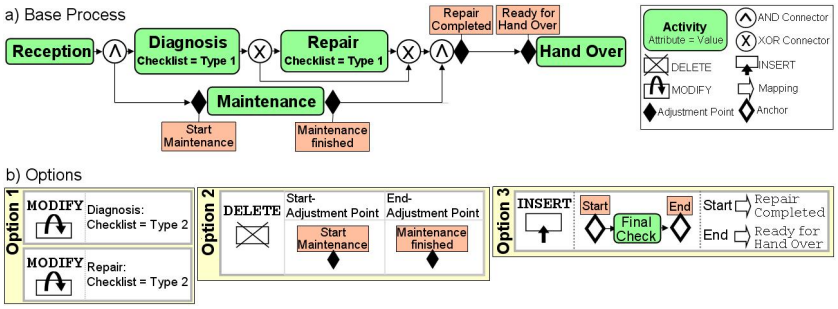

Fig. 3. A base process (a) and related options (b) in Provop

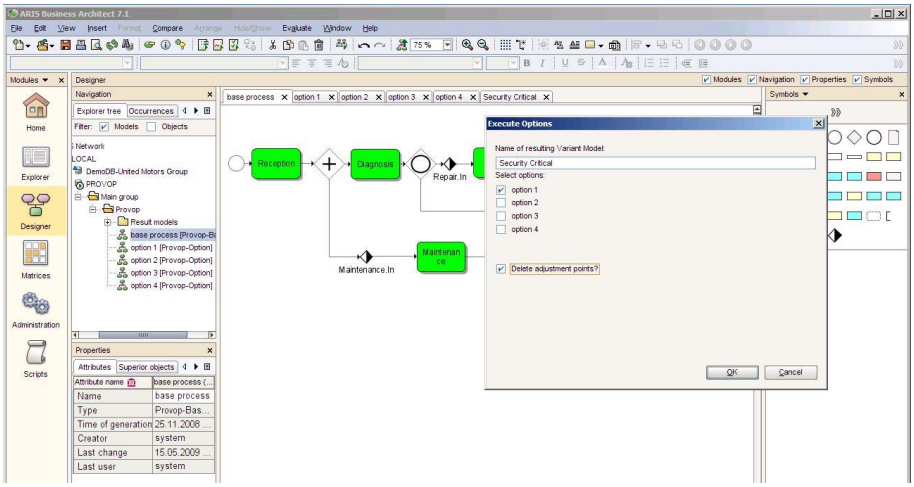

Fig. 4. Configuring a process variant out of options

particular change operation (e.g., adjustment points marking a process fragment to be deleted) are specified using "lanes", which constitute sub-containers of a particular pool and another lane respectively.

A specific ARIS report, which we implemented using ARIS script, realizes the transformation of a base process to a particular process variant. More precisely, for a base process (represented as BPMN model), variant configuration starts with selecting a set of options (cf. Fig. 4). Following this, the change operations of the selected options are applied to the base process. This results in a sound BPMN model, which then represents the configured process variant. In our tool demonstration we give detailed insights into this configuration procedure.

References

1. Hallerbach, A., Bauer, T., Reichert, M.: Guaranteeing soundness of configurable process variants in Provop. In: Proc. 11th IEEE Conference on Commerce and Enterprise Computing (CEC09). (2009)
2. Hallerbach, A., Bauer, T., Reichert, M.: Managing process variants in the process lifecycle. In: Proc. ICEIS'08 Conference. (2008) 154–161
3. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach. Journal of Software Process Improvement and Practice (2009) (accepted for publication).
4. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features - enhancing flexibility in process-aware information systems. Data and Knowledge Engineering **66** (2008) 438–466
5. Hallerbach, A., Bauer, T., Reichert, M.: Context-based configuration of process variants. In: Proc. of the 3rd Int. Workshop on Technologies for Context-Aware Business Process Management (TCoB'08). (2008) 31–40