Faster and better Business Process Modeling with the IBM Pattern-based Process Model Accelerators

Cédric Favre[1], Thomas Gschwind[1], Jana Koehler[1], Wolfgang Kleinöder[1], Alexander Maystrenko[1], Krenar Muhidini[3], Hagen Völzer[1], and Janette Wong[2]

[1] IBM Zurich Research Laboratory, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland
[2] IBM Software Group
[3] Jacobs University Bremen

Abstract. The IBM Pattern-based Process Model Accelerators add a set of patterns, transformations, refactoring operations, and a control-flow analysis feature to IBM WebSphere Business Modeler that make it easy for users to apply common best practices when modeling business processes. As a result, users create and edit higher-quality process models and they benefit from the automation of tedious editing tasks resulting in productivity gains and a more enjoyable user experience. This demo targets participants with interest on faster process modeling approaches leading to higher-quality process models.

1 Introduction

Many tools exist that allow users to model business processes. Traditionally, these tools use a drawing approach where users drag and drop modeling elements on the canvas and connect them by control and data flow. The loose connection between the drawing operations and the actual process modeling leads to the following problems: Manual operations slow down the modeling process and switch the user's attention from the modeling problem to a drawing exercise. Users often lose track of the control-flow that they design when connecting single modeling elements and adding gateways to their process model. As a result, many traditionally created process models contain modeling errors and, for example, cannot be correctly simulated.

We demonstrate the IBM Pattern-based Process Model Accelerators[1] for IBM WebSphere Business Modeler that allow users to compose business process models from patterns and to apply semantically correct change operations to the entire model or selected parts. Although the business process patterns are well-known from the literature, e.g., as workflow patterns [1], there are no modeling tools that would support users in correctly applying patterns. Only the ADEPT 2 editor [2] supports some block-oriented editing patterns. Our accelerators also contain a control-flow analysis feature that detects and displays control-flow errors.

[1] See `http://www.ibm.com/developerworks/websphere/tutorials/0906_gschwind/`.

2 Patterns, Transformations, and Refactoring operations

The accelerators are based on the pattern-based process editing techniques described in [3]. They are beneficial in many process modeling scenarios, e.g., when creating a process model from scratch, refining an existing process model, or correcting errors.

For easy access to the accelerators, we have designed a palette as shown in Figure 1 that provides an icon for each available pattern, transformation, and refactoring operation. An accelerator is applied by optionally selecting some model elements and then clicking on one of the icons in the palette.

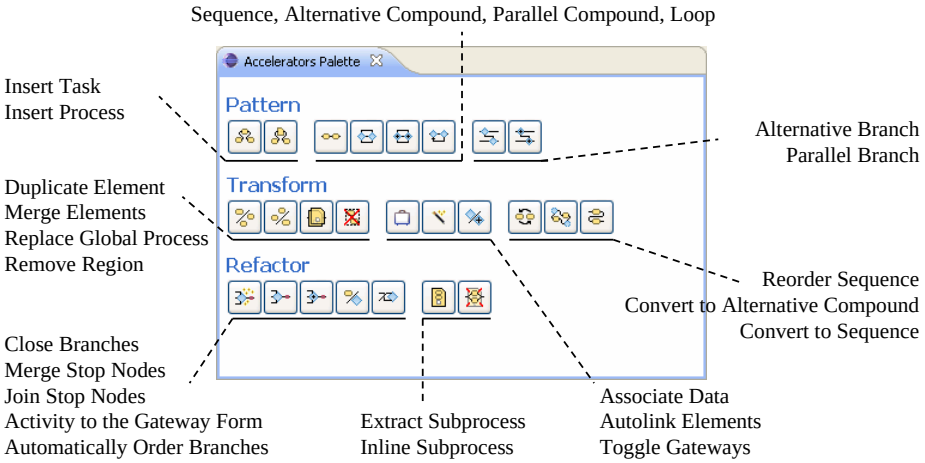


Fig. 1. Accelerators Palette.

Patterns describe frequently used modeling solutions for recurring process scenarios and allow users to model processes by composing basic process fragments. To instantiate a pattern, a wizard is available that contains a picture and description of the pattern as well as fields to enter parameters such as the names of gateways and activities and their data inputs or outputs, see Figure 2. In addition, the Alternative and Parallel Branch patterns allow users to correctly create models with unstructured, i.e., non-block structured control-flow.

Transformations allow users to easily apply a complex change to a process model. A transformation can be understood as a macro-editing operation that combines a sequence of several editing steps that add, delete, or modify elements in the process model. For example, the Autolink Elements transformation adds control-flow connections to elements that are already placed on the canvas following a recommended graphical layout.

Refactoring operations allow users to restructure an existing process model without changing its external behavior and have been inspired by known software refactoring practices, see for example [4]. Refactoring operations are used to improve the readability and to simplify the structure of the process model. For example, the Close Branches operation manipulates the flow of a process model, whereas the Extract and Inline Sub-

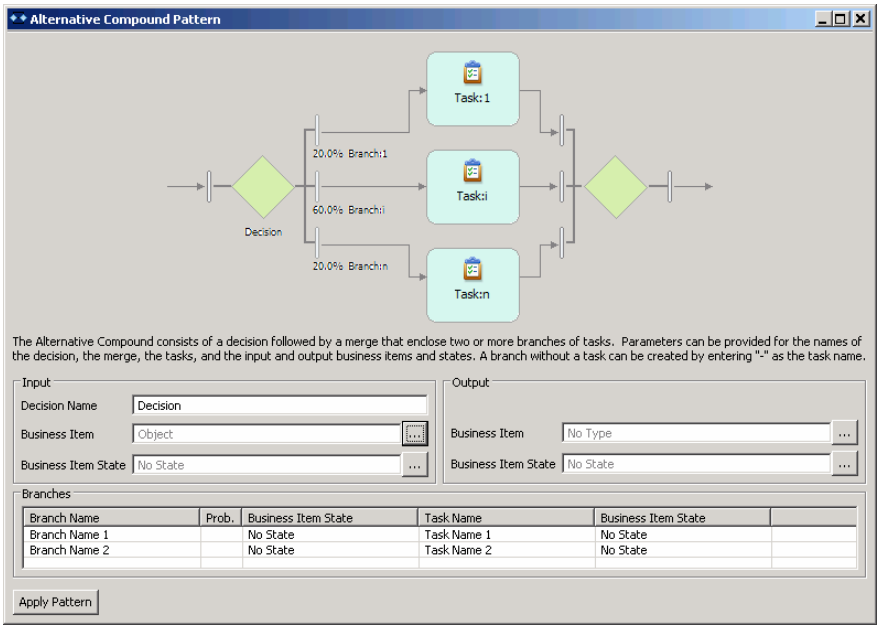


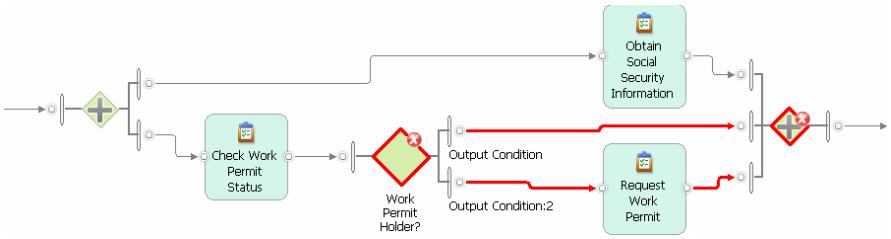
Fig. 2. Wizard of the Alternative Compound Pattern.

process operations allow users to reorganize the model elements across the subprocess hierarchy found in process model.

3 Control-flow analysis

Composing process models using patterns guarantees that process models are free of control-flow errors. However, many traditionally created models contain control-flow errors. As it is likely that users may interleave pattern application with manual editing steps, it is necessary to make it easier to detect and locate control-flow errors. Therefore, the accelerators contain a control-flow analysis component that embeds a soundness checker into a business-user-oriented interface. The checker implements the classical notion of soundness to characterize the correctness of a process model [5] and distinguishes two types of control-flow errors: *deadlock* and *lack of synchronization*. As analysis technique, state-space exploration is used, combined with the parsing technique provided by the Process Structure Tree (PST) [6] that decomposes a process model into smaller fragments that can be analyzed in isolation. A recent case study [7] has shown that models can be checked within a few milliseconds, which allows users to perform an analysis at any time without slowing down the editing process.

The user triggers the analysis for a process or a process catalog. The analysis results are displayed in form of a message indicating the error type, the erroneous process fragment, and, if possible, the cause of the error by some identified model element(s). By clicking on a message, the error is visualized in the process model by highlighting the edges of the process fragment in which the error was localized and marking the identified model element(s), see Figure 3. One run of the analysis can detect several independent control-flow errors within one process model—up to one error per fragment.



Error: Fragment between *Work Permit Holder?* and *Join* contains a deadlock caused by (*Work Permit Holder?*) and detected on (*Join*)

Fig. 3. A deadlock in a process model and the corresponding error message.

Once an error has been identified, transformations can be used to correct the error. By linking some of the transformations to the PST, the behavior of these transformations can be controlled. For example, the Toggle Gateway transformation toggles AND and XOR gateway's logic based on the PST. In an unsound fragment, only a selected gateway will be toggled, whereas in a sound fragment, all gateways will be toggled that are necessary to preserve the fragment's soundness.

4 Conclusion

In this demo, we present the IBM Pattern-based Process Model Accelerators that add a set of patterns, transformations, refactoring operations, and a control-flow analysis feature to the IBM WebSphere Business Modeler tool. By using the accelerators, users move away from traditional process drawing to an approach in which they apply common best practices. Complex process models can be created much faster and are free of control-flow errors. The benefits of an improved process model quality are manifold when simulating processes, translating them into execution language, or executing them directly by a workflow engine.

References

1. Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distrib. Parallel Databases **14**(1) (2003) 5–51
2. Reichert, M., Rinderle, S., Kreher, U., Dadam, P.: Adaptive process management with ADEPT2. In: 21st Int. Conference on Data Engineering, IEEE (2005) 1113–1114
3. Gschwind, T., Koehler, J., Wong, J.: Applying patterns during business process modeling. In: BPM-08. Volume 5240., Springer LNCS (2008) 4–19
4. Fowler, M.: Refactoring — Improving the Design of Existing Code. Addison-Wesley (2008)
5. Aalst, W.M.P.v.d., Hirnschall, A., Verbeek, H.M.W.: An alternative way to analyze workflow graphs. In: CAiSE-02. Volume 2348., Springer LNCS (2002) 535–552
6. Vanhatalo, J., Völzer, H., Koehler, J.: The refined process structure tree. In: BPM-08. Volume 5240., Springer LNCS (2008) 100–115
7. Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Instantaneous soundness checking of industrial business process models. In: BPM-09. Accepted.