# Case Study: BPMN to BPEL Model Transformation

Marlon Dumas

University of Tartu, Estonia
marlon.dumas@ut.ee

**Abstract.** This case study considers the definition of model transformations between two languages for business process modeling, namely BPMN and BPEL. The model transformations should achieve four evaluation criteria: completeness, correctness, readability and reversibility.
**Keywords:** Model transformation, BPMN, BPEL

## 1 Description

The purpose of this case study is to define a model transformation between two standard languages for business process modeling: the Business Process Modeling Notation (BPMN) [6] and the Business Process Execution Language (BPEL) [4].

BPMN is a standard notation for modeling business processes for the purposes of business analysis. The target users of this notation are business analysts. BPEL on the other hand is intended as a notation to define executable processes, that is, processes that can be loaded into an execution engine which will then coordinate the activities specified in the process definitions.

In order to facilitate handovers between business process analysts and developers, several commercial tools (e.g. Oracle BPA, ARIS, Intalio, ITP-Commerce Process Modeler) promote an approach to process-oriented software development in which BPMN process models are transformed into BPEL process definitions. These transformations can be one-way transformations (BPMN-to-BPEL only) or ideally two-way (reversible) transformations. The problem of defining these transformations turns out to be rather complex because of inherent differences between these two languages: BPMN process models are graph-oriented (with only minor topological restrictions), while BPEL process definitions are block-structured. The definition of BPMN-to-BPEL model transformations is straightforward for some classes of BPMN models, but less obvious for others.

## 2 Meta-models and Transformation

The meta-models of the subsets of BPMN and BPEL that we consider in this case study are given in Figure 1 and Figure 2 respectively. For full details about BPMN, see [6] and for full details about BPEL, see [4].

The definition of transformations from BPMN to BPEL has been investigated by several authors [2, 5]. In this case study, we consider the transformations

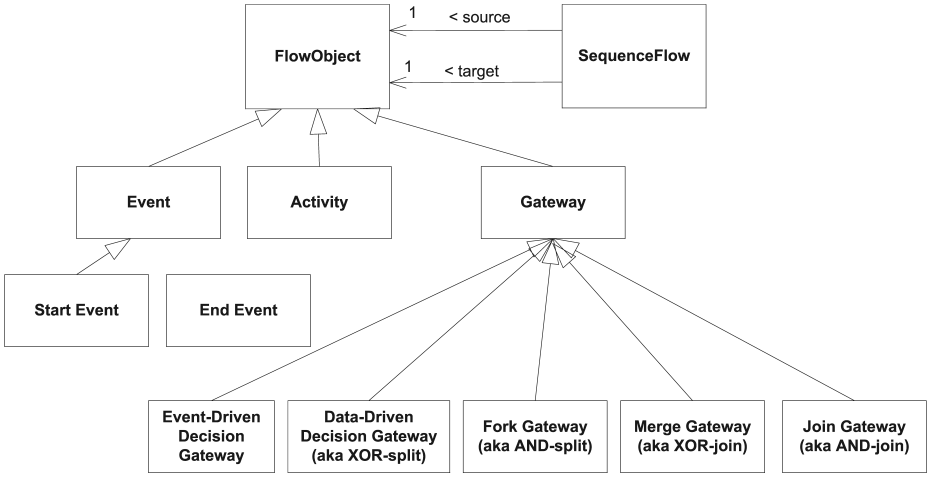


**Fig. 1.** Relevant fragment of the BPMN meta-model

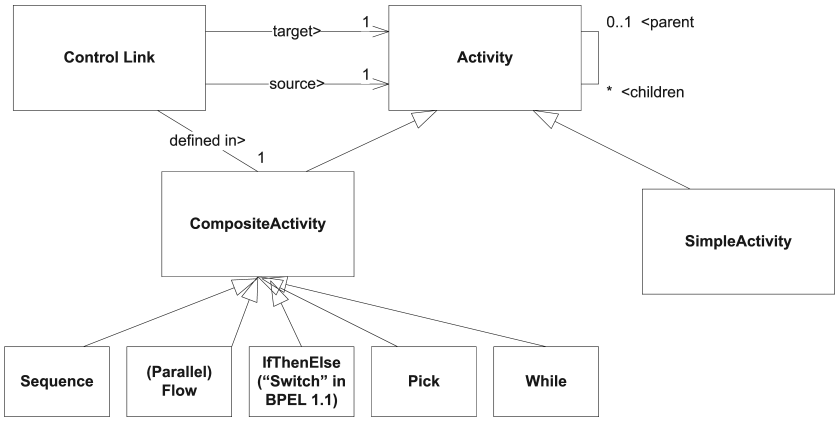


**Fig. 2.** Relevant fragment of the BPEL meta-model

defined in [7, 8]. Two implementations of these transformation exist. One is the BPMN2BPEL command-line tool.[1] The second implementation is integrated as an export function in the Oryx online BPMN process model editor.[2]. These are two distinct implementations, with the latter one being more extensive in its coverage of the BPMN notation than the former one.

One of the key issues of the BPMN to BPEL transformation is that it requires one to identify complex patterns in the input model, in particular single-entry single-exit (SESE) regions. The problem of identifying all SESE regions in a graph has been discussed by Johnson et al. [3].

## 3 Tasks and Variations

The task is to define a transformation from the subset of BPMN represented by the above meta-model into BPEL, using a graph transformation approach. We consider three variants of this task corresponding to different levels of completeness of the transformation:

1. Transformation restricted to structured process models as defined in [8].
2. Transformation restricted to structured and quasi-structured process models as defined in [8].
3. Transformation covering structured and quasi-structured process models, as well as synchronizing process models as defined in [7]

In addition to completeness, the definition of the model transformation should consider three evaluation criteria: correctness, readability and reversibility.

Firstly, the transformation should preserve the execution semantics of the BPMN process model. Specifically, the BPEL process definition obtained from the transformation should be weak-bisimulation equivalent to the original BPMN process model.[3] Since BPMN and BPEL are not formally defined in their respective standard specifications, we can not formally define weak-bisimulation equivalence on them. However, transformations exist from BPMN to Petri nets [1] and from BPEL to Petri nets [9]. So if this was necessary, equivalence can be checked on the corresponding Petri nets.

Secondly, the BPEL process definitions produced by the transformation should be "human-readable". The mappings presented in [7, 8] have been designed with this criterion in mind, but improvements are possible. Previous work suggests that when defining a BPMN-to-BPEL model transformation, there is a tradeoff between covering larger classes of BPMN process models and producing human-readable BPEL process definitions. For the purposes of this case study, readability of the BPEL process definition is left as a subjective criterion. As a guideline, the use of BPEL's block-structured control-flow constructs (sequence, while, if-then-else and pick) should be preferred over using control links or using other

[1] Available at `http://www.bpm.fit.qut.edu.au/projects/babel/tools/`

[2] Available at `http://www.bpel4chor.org/editor/`. The use of this editor requires the Firefox browser (version 2.0).

[3] To the best of our knowledge, this is the case of the transformations in [7, 8].

control-flow constructs such as event handlers, since block-structured constructs are generally more readable when displayed in existing BPEL editors.

Finally, the BPMN-to-BPEL transformation should preferably be accompanied by a reverse transformation. In other words, there should ideally be two transformations: one from BPMN to BPEL, and another from BPEL to BPMN such that if a BPMN model $B$ is transformed into a BPEL definition $E$ and then $E$ is transformed back into BPMN, the resulting BPMN model is identical (or as similar as possible) to $B$. An alternative to making the transformation reversible would be to provide, in addition to a model transformation from BPMN to BPEL, a mechanism for bidirectional model synchronization allowing users to keep the BPMN process model and its corresponding BPEL process definition synchronized when either or both of them undergo changes.

## References

1. R. Dijkman, M. Dumas, and C. Ouyang. Semantics and analysis of business process models in BPMN. *Information and Software Technology*, 2008. To appear. A technical report version is available via `http://eprints.qut.edu.au/archive/00007115/`.
2. R. Hauser and J. Koehler. Compiling process graphs into executable code. In *Proceedings of the 3rd International Conference on Generative Programming and Component Engineering (GPCE 2004), Vancouver, Canada*, pages 24–28. Springer, October 2004.
3. R. Johnson, D. Pearson, and K. Pingali. The Program Structure Tree: Computing control regions in linear time. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 171–185, Orlando FL, USA, June 1994. ACM Press.
4. D. Jordan and J. Evdemon. *Web Services Business Process Execution Language Version 2.0.* Committee Specification. OASIS WS-BPEL TC, January 2007. Available via `http://www.oasis-open.org/committees/download.php/22475/wsbpel-v2.0-CS01.pdf`.
5. J. Mendling, K.B. Lassen, and U. Zdun. Transformation strategies between block-oriented and graph-oriented process modelling languages. In F. Lehner, H. Nösekabel, and P. Kleinschmidt, editors, *Multikonferenz Wirtschaftsinformatik 2006. Band 2*, pages 297–312. GITO-Verlag, Berlin, Germany, 2006.
6. OMG. *Business Process Modeling Notation (BPMN) Version 1.0.* OMG Final Adopted Specification. OMG, February 2006. Available via `http://www.bpmn.org/`.
7. C. Ouyang, W.M.P. van der Aalst, M. Dumas, and A.H.M. ter Hofstede. Translating BPMN to BPEL. Technical Report BPM-06-02, BPMcenter.org, January 2006. Initial version available at `http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-02.pdf`. Revised version available at: `http://math.ut.ee/~dumas/downloads/BPMN2BPELRevised.pdf`.
8. C. Ouyang, M. Dumas, A.H.M. ter Hofstede, and W.M.P. van der Aalst. Pattern-based translation of bpmn process models to bpel web services. *International Journal of Web Services Research*, 5(1), January 2008. Available at: http://is.tm.tue.nl/staff/wvdaalst/publications/z9.pdf.
9. C. Ouyang, H.M.W. Verbeek, W.M.P. van der Aalst, S. Breutel, M. Dumas, and A.H.M. ter Hofstede. Formal semantics and analysis of control flow in WS-BPEL. *Science of Computer Programming*, 67(2-3):162–198, 2007.