Towards Synthesizing BPMN2 Test Models for Model Differencing Tools

Pit Pietsch and Christian Wurmbach
Software Engineering Group
University of Siegen, Germany
{pietsch, cwurmbach}@informatik.uni-siegen.de

ABSTRACT
In the last decade Model Driven Engineering (MDE) became a central development paradigm in many application domains. Thus, many tools and methods are introduced to the community which subsequently have to be tested and evaluated. Unfortunately test models are not or only scarcely available for many of these domains. The usual approach in these cases is to use synthetically generated test models. Generally, synthetic test models should resemble real models and their editing processes as closely as possible. Hence, it is necessary to identify the typical edit behaviour of modelers, i.e. the edit profile, for a model type first. This paper discusses the first steps in the identification of an edit profile for BPMN2 models. To this end we selected 32 histories with a total of 855 BPMN2 models from the BPMAI database and converted these into the eclipse implementation of the BPMN2 standard. This paper states some of the problems we encountered selecting and processing the sample set. The converted Eclipse BPMN2 models are available for download. In our future work we will use this sample set to identify typical edit behaviour in the domain of BPMN2.

1. MOTIVATION
Model-Driven Development (MDD) has become a central development paradigm in many application domains. In MDD models are the central artifacts of development and replace source code. The models themselves are collaboratively and concurrently developed by teams of modelers. Hence, the same problems as for the version management of source code arise for models, too. Because of this model versioning related tools were especially in the focus of research in the last couple of years. Model comparison\(^1\), i.e. the identification of common elements in two models as well as the edit operations which transformed the first model in the second, plays a very important role within model versioning tools. Model comparison is essential for many advanced model versioning functionality, e.g. difference visualisation, 2-way or 3-way merge of models and model patching. Unfortunately, model comparison tools have to be adapted for each model type to compute high quality results [4].

For Business Process Model and Notation 2.0 (BPMN2) diagrams a configuration for the SiDiff model differencing framework [4] was introduced in [6]. While this initial configuration produces promising results, we were not able to further refine it due to the lack of BPMN2 models for testing. We want to overcome this scarcity by generating synthetic test models with the SiDiff Model Generator (SMG) [7]. The SMG can, if the edit profile for a domain is known, generate pairs or sequence of models which mimic real models as well as real editing processes [8, 10]. Hence, our primary goal is to identify the typical edit profile for BPMN2 diagrams.

2. APPROACH IN A NUTSHELL
Figure 1 shows the standard approach [10] for the identification of edit profiles. Consecutive revisions in representative model histories are compared and the low-level differences are computed. Low-level differences are based on simple edit operations, i.e. creating or deleting a model element, update of attribute values for a given model elements and the move of a model element. These low-level differences are often hard to understand for developers, because they are used to modify the models based on higher level edit operations. Such high-level operations, e.g. refactorings like pullUpAttribute are usually comprised of several low-level operations. Hence, we want to lift the low-level differences so that we are able to identify the edit behaviour based on the abstraction level the modelers work on. This is done by application of the method described in [5]. Finally, based on the lifted differences we will analyse which statistical models best describe the actual changes. These models represent the edit profile and are used to configure the SMG accordingly.

3. SAMPLE SET SELECTION
When searching for suitable process models for the sample data set we discovered the BPM Academic Initiative (BPMAI) [1]. The BPMAI is a platform which offers free comprehensive teaching materials to foster the use of process models in university courses. Furthermore, an industry supported online modeling editor is available. All models created with this modeler are automatically stored and are available for research in the context of process models.

\(^1\)See [9] for a survey on the state-of-the-art in model comparison.
For BPMN2 there are 66812 models available. Because of the large quantity and the fact that many of the models were trivial, i.e. they represent only very simple process models, oftentimes consisting of only very few elements, we decided to limit the selection. Hence we decided to apply the following constraints to identify models which can be considered as representative. We limited the sample set to histories which have at least 10 revisions (Constraint A). Furthermore, at least one model in the history must consists out of more than 250 elements and 25 different element types (Constraint B). All model histories which fulfilled these requirements where then checked by an domain whether or not they can be considered as typical process models (Constraint C). Based on this procedure we identified 32 histories with a total of 855 models as the representative sample data set.

4. SAMPLE SET TRANSFORMATION
The models from the BPMIAI data set are stored based on an implementation of the BPMN2 standard for the PromniCAT framework [3]. PromniCAT is a collection and analysis framework for process models. Unfortunately, this format cannot be directly processed by our tools, which are based on the Eclipse BPMN2 implementation of the standard provided by the BPMN2 Modeler [2]. Therefore we had to write a transformation which takes one or more BPMN2A model(s) as output. The transformation is straightforward and almost all of the elements from the PromniCAT representation could directly be mapped to the corresponding element types in the Ecore BPMN2 representation. In some cases the PromniCAT implementation deviates from the standard, though. For example it is still possible to have ReferenceTasks in PromniCAT BPMN2 diagrams, although this element type was only available in until the 1.2 revision of the standard. Also few element types, e.g. RuleEvent and ITSystems, are used in PromniCAT which are not defined in the standard. These missing types are mapped to other element types from the standard and are specifically marked, e.g. by their names.

5. CONCLUSION AND FUTURE WORK
Now that the work on the transformation is finished, we will compared the models based on the current SiDiff configuration. The results will be checked by an domain expert for plausibility. Furthermore we are currently specifying about 20 refactorings for BPMN diagrams which will then be used to lift the low level differences. The collected data is then used to find the mathematical model which fits the findings, i.e. the edit profile for BPMN2 models. This information is then used to configure the SMG and to produce realistic benchmarks for model comparison tools.

6. REFERENCES