Simulation and Formal Analysis of Workflow Models Using Model Transformations[1]

Máté Kovács

Today major organizations tend to use more and more IT infrastructure that include the use of workflow execution engines as well. The *workflow* coordinates the activities inside the company, this way it becomes crucial concerning the correct functionality.

There are several high level executable languages for workflow implementation for example: BPEL (Business Process Execution Language). However, the testing of such workflows is problematic. The engine may be in contact with several independent databases that may be in the possession of different organizations. The rollback of the effects of each test run and the establishment of a test environment with all the data is equally expensive. My proposal aims to solve this problem with the formal model checking of workflows.

The BPEL is a semiformal notation. In order to examine its properties with mathematical accuracy, a workflow has to be transformed into an exact formalism. There are several approaches that use Petri nets or nondeterministic automata for this purpose.

I chose *dataflow networks* as the model of workflows. Dataflow networks consist of data processing nodes that are finite state automata interconnected with data conveyer channels. The formalism of dataflow networks gives a *verifiable semantics* to workflows.

In my work [1] I discuss a method, which allows us to carry out the formal analysis of a workflow implemented in BPEL without human intervention. This method relies on graph-transformations that are executed by the Viatra (Visual Automated Model Transformations). The first step (i) is to transform the BPEL model into a dataflow network model, which can be done in a deterministic way, due to the properties of the workflow description. The second transformation (ii) maps the dataflow into a Promela (Process Meta Language) model. The third (iii) step is to automatically generate the real Pomela program. The requirements set against the Promela code have to be formulated as Linear Temporal Logical expressions (LTL) that can be evaluated. This way the dynamic properties of a BPEL model can be formally checked.

The dataflow network model of workflows makes fault simulation also possible. With a small extension of the BPEL language *fault injector activities* are introduced. LTL expressions may be evaluated questioning whether the fault of one activity affects the other. This way the *error confinement region* of a faulty activity can be determined, and it can be verified whether a possible planned redundancy in the workflow model reached its goal.

There are general properties that a workflow definitely has to meet. The concrete LTL expressions, which are specific to the model, can be generated from the general requirements.

If the evaluation of a specific LTL expression results in failure, the sequence of events leading to the counterexample is also presented. The *back annotation* of such event sequence to the BPEL domain is a future goal.

References

[1] M. Kovács and L. Gönczy. Simulation and Formal Analysis of Workflow Models, In *Proceedings of 5th International Workshop on Graph Transformation and Visual Modeling Techniques* (GT-VMT), ENTCS, pp. 215-225, Vienna, Austria, 2006.

[1]This work was partially supported by the SENSORIA European project (IST-3-016004).