Constraint-based analysis of business process models

Changizi, B.

8 Conclusion

Despite long-term efforts, analyzing business processes is still a challenge. Creating tools for analyzing business processes requires expressing the behavior of the processes in an accurate way. Most of the business process management notations, particularly Business Process Model and Notation (BPMN), are based on Petri nets.

While Petri nets can be used to automate process analysis, they are not compositional. This makes analyzing the behavior of large and complex models based on Petri nets challenging.

The Reo coordination language is an alternative theory to Petri nets that has been used to formalize semantics of BPMN. Reo has a compositional nature, which enables adding semantic models for individual components to the semantic models of existing processes.

In this dissertation, we used the Reo coordination language to capture the behavior of BPMN processes. We presented an automated mapping of business process models expressed in BPMN 2 to Reo networks in order to create the possibility of using various types of analysis on business process models. Our mapping takes data into account. Thus, it enables verification of data flow. We not only deal with basic BPMN 2 constructs, but also with compound elements such as transactions and exception handling. Formalizing the behavior of these elements requires modeling priority.

Reo is an extensible language that comes with various formal semantic models. This makes it possible to perform different kinds of analysis by focusing on specific

behavioral aspects of a given network. However, there is a gap between the behavior that each of the semantics can express. This can introduce incompatibilities among these operational models. In addition, these formal semantics are computed using their own specialized algorithms, which are directly implemented.

Such algorithms are computationally expensive. As a result, the Reo models (and consequently business models) whose operational semantics can be efficiently calculated are limited to those of relatively small size.

Each of these formal semantics constrain the possible I/O operations through the nodes to those allowed by the semantics. Therefore, we convert the problem of finding behaviors accepted by a given semantic model into a constraint satisfaction problem for which many efficient supporting tools exist.

We developed a unified constraint-based framework to compute formal semantics of a Reo network given the semantics of its parts in a compositional fashion. Since we have included various existing formal semantics of Reo in our framework, behavior specifications that are considered invalid according to any of these formal semantics are ruled out. The tool we implemented to realize this framework relies on constraint solvers. Therefore, it benefits from the advances in the field of constraint solving.

Within this framework, the behavior of a Reo construct specified by a given semantics model is expressed in terms of constraints. In order to obtain the semantics of the whole Reo connector, the constraints of its constructs are concatenated. The framework replaces data constraints with new binary predicates that represent the logical value of the data constraint. The final constraint is then converted to the acceptable format for an off-the-shelf constraint solver.

After the constraint solver finds the solutions, the solutions are mapped back to the predicates. The data constraints and the value of their representative predicate are sent to a numeric constraint solver that treats the data symbolically. This way instead of obtaining distinct possible values for each variable denoting a data-item, we have a range of values, which is a more compact representation. We compared the performance of our approach to the existing ways of computing the formal semantics of Reo.

We presented a constraint-based approach for calculating priority-aware semantics of Reo models. This approach has been integrated into the mentioned constraint-based framework as the first tool support for priority in Reo. Similarly, this approach benefits from the shift of paradigm from custom direct implementation to using tools available in the well researched area of constraint solving. We not only provide a way to model the binary notion of priority in Reo, but also we deal with numeric priority. We demonstrated the application of our toolchain by analyzing a BPMN process that could not be analyzed previously.

A limitation of our implemented toolchain is that it relies on the external BPMN modeling tools to create the BPMN process to be analyzed. Since not all BPMN tools support export the BPMN models in our expected format, the choice of BPMN editor compatible with our tool set is limited.

As our future work, we plan to expand our constraint-based semantics framework to include other formal semantics of Reo, for instance, those that incorporate stochastic and quantitative aspects of the behavior of Reo circuits. In addition, we plan to extend our constraint-based framework to generate data to be used for

simulation and testing purposes.