Model based Optimization and Verification of IT Systems

Szilvia Gyapay

As the proper services delivered by IT systems became more and more crucial to production and the life of the society their correct and efficient operation has to be proven already during the design. Validation and verification methods are known to assure the correctness of the services while optimization may serve both to the proof of the quantitative performance characteristics of a system by estimating its quantitative boundaries and to minimize operation costs. In order to avoid costly redesigns, the problems of fulfillment of the requirements have to be addressed already during the early design phases. Recently, system designers use the Unified Modeling Language (UML) that became the standard modeling language since it provides a semiformal, concise description of complex systems for modeling both its static and dynamic behavior. UML can be extended to incorporate quantitative measures, requirements, and constraints.

On the one hand, UML is a proper means for system and requirements modeling but the analysis itself has to be carried out by using some mathematical model analysis tool. Recent research efforts aim at automated transformation between UML models and mathematical analysis tools [3].

One of the most challenging problems in mathematical analysis of IT systems is the simultaneous analysis of the dynamical behavior of the system and its impact to the quantitative measures, as this meets a combination of a mathematical paradigm describing the control logic of the application and the quantitative impact of it. We propose to use the combination of Petri nets as general purpose logic modeling paradigm and Process Network Synthesis as an optimization framework [1]. Our objective is to derive from the UML specification of the application and to estimate the worst case characteristics of a system upon temporal constraints on its operation sequence.

- At first, an automated, attributed graph grammars based transformation [2] maps the UML models into Petri nets.
- Secondly, all firing sequences representing possible operations (trajectories from an initial state to a given state), that satisfy specific, user defined constraints, are estimated as a basis spanning the state space of the feasible solutions to the optimization objective.
- The next step is to compute a candidate trajectory that represents an optimal operation by combining trajectories generated in the previous step.
- In order to ensure the executability (fireability) of the candidate trajectory, post filtering is executed by model checking (exhaustive simulation).

The new contribution of the current presentation compared to previous ones is the introduction of a multiple level set of feasibility check of algorithms and the benchmark based analysis of the efficiency of the approach.

References

[1] S. Gyapay and A. Pataricza. A combination of petri nets and process network synthesis. In 2003 IEEE International Conference on Systems, Man Cybernetics, Invited Sessions / Track on "Petri Nets and Discrete Event Systems", pages 1167–1174, Washington, D.C., USA, October 58 2003.

[2] G. Rozenberg, editor. Handbook of Graph Grammars and Computing by Graph Transformations, vol. 1: Foundations. World Scientific, 1997.

[3] D. Varro and A. Pataricza. VPM: A visual, precise and multilevel metamodeling framework for describing mathematical domains and UML (The Mathematics of Metamodeling is Metamodeling Mathematics). Journal of Software and Systems Modelling, 2(3):187–210, 2003.