Verification of Reconfiguration Mechanisms in Service-Oriented[1]

László Gönczy

As the automated integration of heterogeneous software environments becomes widespread, there is a growing demand for resilient software architectures. The Service-Oriented Architecture is an emerging paradigm in this field, however, in its present form, it does not cover the fault-tolerant aspects and no verifiable system reconfiguration mechanisms are modeled.

As the configuration of the underlying services may often change, (e.g. consider a network of mobile services), the system must be able to react to effects of the changes of the environment by means of *dynamic reconfiguration*. On the other hand, certain properties (such as the availability of the minimal number of service instances of a given service type, the presence of a particular service or some quantitative requirements) must be guaranteed during the entire lifetime of the system. Therefore, the aim of my ongoing research is twofold:

First, the *components* of a typical Service Oriented Architecture (services, ports, messages, etc.), their non-functional *properties* (such as the guaranteed response time or the acknowledgement options), the *reconfiguration* mechanisms (e.g. searching for and invoking a new service if a call fails or the response time decreases below a certain limit), and the *fault model* (e.g. a service crashes or becomes overloaded) are described in a high-level model while the underlying technology, for instance, SOAP messages or Grid technologies will remain hidden [1]. The basis of the followed approach is discussed in [2].

Second, I implemented a system for the *verification* of reconfiguration algorithms. Based on the high-level, technology-independent system description, i) the state space of the model will be investigated by *reachability analysis* to generate the set of the possible succeeding system configurations and ii) the fulfillment of the requirements against these configurations (and the transient states) are verified. During this analysis, a graph transformation tool with a model checking support can be used, like Groove [3]. The most important future research task is the investigation of the applicability model-based synthesis of Service-Oriented Architectures and service configurations.

References

[1] L. Gönczy and D. Varró. Modeling of Reliable Messaging in Service Oriented Architectures, In *Proc. Intern. Workshop on Web Service Modeling and Testing (WS-MATE 2006)*, To appear.

[2] L. Baresi, R. Heckel, S. Thöne, and D. Varró. Style-Based Modeling and Refinement of Service-Oriented Architectures - A graph transformation-based approach, To appear in *Software and Systems Modeling* - Springer-Verlag, 2006.

[3] Graphs for Object-Oriented Verification (GROOVE project), `http://groove.sourceforge.net/groove-index.html`

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