Dependability Modeling of Web Service Flows

László Gönczy[1], András Pataricza[1], Felicita Di Giandomenico[2,] Silvano Chiaradonna[2] and Andrea Bondavalli[2]

[1] DMIS, Budapest University of Technology and Economics
Magyar Tudósok krt. 2. H-1117, Budapest, Hungary
{gonczy, pataric}@mit.bme.hu
[2] ISTI-CNR, Pisa Research Area - Via Alfieri 1 - I56010 Ghezzano, Pisa, Italy
{a.bondavalli, f.digiandomenico,
silvano.chiaradonna}@isti.cnr.it

1 Introduction

The trend of service-based integration of heterogeneous, loosely-coupled systems raises the problem of dependability analysis of process descriptions. These integrated systems typically implement workflows built of distributed services, over which the provider of the main service does not have a complete control. Therefore, the non-functional parameters of the basic services have to be described by Service Level Agreements, so that the main service provider can analyse the different system configurations (i.e. processes built of different external services or of the same services with different SLAs).

In our research we investigate the possibilities of analysing Web service flows with given SLAs by generating a dependability model of the system as a Multiple-Phased System.

2 Dependability Modelling of Web Service Flows

Web service flows can be described by several workflow languages (such as BPEL designed specially for this kind of distributed workflows). These languages typically support the integration of distributed enterprise systems, but lack of describing non-functional parameters. Such parameters can relate to performance, security, dependability, etc. Our research aims at building a dependability model of the process based on the SLAs. A typical dependability parameter of an SLA is availability. As traditional workflow descriptions do not contain dependability parameters, they have to be extended to capture dependability parameters.

Multiple Phased Systems (MPS) [1] is a class of systems the behavior of which shows phases during their operational life. These phases may have different performance and dependability requirements, such as throughput, response time, availability, etc. Execution of a given sequence of tasks may depend on the results of previous phases, and the so-called mission goal may change over time. Previously, MPS tools were used to describe mission critical systems such as reactor protection systems,

spaceships and GPRS cells. Our research aim is to apply this kind of system description for flow models of Web service systems.

The concrete modeling tool chosen was DEEM (Dependability Evaluation of Multiple Phases Systems) which uses Deterministic and Stochastic Petri Nets as modeling paradigm and a model solution algorithm based on Markov Regenerative Processes.

3 Necessary Extensions of Business Process Descriptions

Web service flows can be considered a special subclass of Phased Mission Systems because: i) their operational life is built of distinct steps, ii) these steps need different resources and have different dependability requirements based on Service Level Agreements, and iii) the execution of basic steps depend on the results of previous tasks.

There are four possibilities to combine BPM and MPS paradigms:

- Describe the process flow in BPM and transform it to some MPS representation.
- Describe the process in a meta-language and transform this description to BPM (for implementation) and MPS (for dependability analysis).
- Describe the process in MPS and transform it to BPM.
- Describe the process in both modeling paradigms.

We chose to describe the process in BPM and then perform model transformation.The main reasons for this were the following:

- No additional meta-language has to be developed.
- Wide-accepted and easy-to-learn modeling tools can be used so process engineers do not have to learn Petri nets' semantics.
- An implementation "skeleton" can be easily generated using concrete Web services as process activity implementation. For this, the BPM representation has to be transformed into a BPEL model which can be done by commercial tools.
- Automatic (or semi-automatic) transformations can guarantee that the two representations of the process are equivalent.

The model transformation needs an engine which helps us to implement graph transformations (based on rules applied to certain graph patterns). The VIATRA 2 framework [2] –developed at our Department– is constructed to help the design of such transformations. The VIATRA 2 also has a parser for BPM models. This way a lot of effort can be spared at developing transformations.

References

1. A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and I. Mura, "Dependability modeling and evaluation of multiple-phased systems, using DEEM," IEEE Transactions on Reliability, 2004.
2. D. Varró, G. Varró, A. Pataricza, "Designing the Automatic Transformation of Visual Languages," Science of Computer Programming, 44:205-227 2002.