Dependability Modeling of Fault-Tolerant Systems Using Aspect-Oriented Modeling Techniques

Péter Domokos and István Majzik

Dependability modelling and analysis is useful for the understanding and assessment of the system in all phases of its life cycle. The two main quantitative dependability attributes of interest for a designer are availability (the delivery of correct service with respect to the alternation of correct and incorrect service) and reliability (the continuous delivery of the correct service). During design phases, dependability models allow to compare different architectural and design solutions, to select the most suitable one, and to highlight dependability problems in the design. Dependability models are mathematical models which describe the failure and repair processes of individual components of the system and the error propagation between the components. This model is used to determine when the failure of individual components leads to a system level failure, that is, to incorrect service.

The analysis is based on the UML model of the system. UML (Unified Modeling Language) is a modeling language which supports the entire design process: the requirement analysis is aided with activity diagrams and statecharts, the system architecture design is supported with class diagrams and object diagrams, the implementation is supported with statecharts and sequence diagrams. The dependability model is automatically derived from the UML model of the system as described in [1],[2].

In the early phase of the design process, architectural models are typically created (eg. class diagrams). This model is extended with the dependability parameters of the components as tagged values (eg. fault occurrence, error latency, ratio of permanent faults, repair delay). The dependability model is automatically derived from the class diagrams by graph transformation. In order to increase the reliability and the availability of dependable systems, fault tolerance structures are usually applied, eg. TMR (Triple Modular Redundancy) in hardware or N-version programming in software, where N different modules solve the same problem and the results are compared by a voter.

The redundancy management forms an independent aspect: the core functionality of the system forms a concern, while the fault-tolerance management layer provides another concern. Aspect-oriented programming is a new approach for separating these independent concerns [3]. Aspect-oriented programming proposes that each concern should be implemented independently and that the final system be derived by weaving these aspects into a single system based on the weaving rules. Such a weaving rule may be for example that a so-called advice (eg. logging) should be executed before the calling of the public functions of a given class. We aim at using aspect-oriented modeling and aspect-oriented programming for the modeling and implementation of fault tolerance structures. We will derive the dependability models from architectural UML models using aspect-oriented extensions for modeling redundancy techniques.

References

[1] I. Majzik, A. Pataricza, and A. Bondavalli. Stochastic dependability analysis of system architecture based on UML models. In Rogerio de Lemos, Cristina Gacek, and Alexander Romanovsky, editors, Architecting Dependable Systems, volume LNCS-2677, pages 219-244. Springer, 2003.

[2] P. Domokos, I. Majzik: From UML class diagrams to timed Petri nets. Proceedings of the 11th PhD Mini-Symposium, Budapest University of Technology and Economics, Febr. 3-4, 2004.

[3] T. Elrad, M. Aksits, G. Kiczales, K. Lieberherr, H. Ossher. Discussing aspects of AOP. Communications of the ACM, volume 44, issue 10, pages 33-38, October 2001.