Software Architectures of Dependable Systems: From Closed to Open Systems

Valérie Issarny
INRIA, UR Rocquencourt
Domaine de Voluceau - B.P. 105 - 78153 Le Chesnay France
Valerie.Issarny@inria.fr

INTRODUCTION

Work in the software architecture domain primarily focuses on the standard (as opposed to exceptional) behavior of the software system. However, it is crucial from the perspective of software system robustness to also account for failure occurrences. The next section gives an overview of our past work towards assisting architecting of dependable distributed systems. It is then followed by a discussion on our current and future research work towards addressing dependability requirements of *open* distributed systems, which are expected to become a major class of future distributed systems.

AIDING THE ARCHITECTING OF DEPENDABLE SYSTEMS

Failures may be handled through the integration within the system architecture of components and connectors that provide fault tolerance capabilities. Practically, this means that failures are handled by an underlying fault-tolerance mechanism (e.g., transparent replication management) at the middleware level. Such fault tolerance support must further be coupled with software fault tolerance that relies at least on an exception handling mechanism, which enables the software developer to specify the actions to be undertaken under the occurrence of application-specific and underlying runtime exceptions. We have then carried out research in the two following complementary directions towards assisting architecting of dependable systems.

Systematic aid in the development of middleware architectures for dependable systems: The use of middleware is the current practice for developing distributed systems. Developers compose reusable services provided by proprietary or standard middleware infrastructures to deal with non-functional requirements. However, developers still have to design and implement middleware architectures combining available services in a way that best fits the application's requirements. In order to ease this task, we have developed an environment that provides [1]: (i) an ADL for modeling middleware architectures, (ii) a repository populated with architectural descriptions of middleware services, and (iii) automated support for composing middleware architectures out of available services according to target non-functional properties, and for quantitatively assessing the composed architectures in terms of performance and reliability.

Architecture-based exception handling: As previously raised, it is necessary to complement fault-tolerance support provided by the underlying middleware architecture, with support for software fault tolerance so as to enable application-specific fault-tolerance. We have thus proposed a solution to architecture-based exception handling [2], which complements exception handling implemented within components and connectors. Our solution lies in: (i) extending the ADL so as to enable the specification of required changes to the architecture in the presence of failures, and (ii) associated runtime support for enabling resulting dynamic reconfigurations.

FUTURE RESEARCH DIRECTIONS

The above results have been proven successful for assisting the architecting of robust distributed systems that are closed, i.e., systems whose components depend on a single administrative domain and are known at design time. However, future distributed systems will increasingly be open, which raises new issues for making them dependable. In this context, we are in particular undertaking research in the following directions: (i) Architecting open distributed systems in a way that accounts for mobility, which requires support for the dynamic composition and quality assessment of architecture instances; and (ii) Design of fault-tolerance mechanisms for open distributed systems considering that the systems span multiple administrative domains and hence cannot accommodate locking-based solutions as, e.g., enforced by transactional processing [3].

REFERENCES

[1] Issarny, V., Kloukinas, C. and Zarras, A. Systematic Aid in the Development of Middleware Architectures. *Communications of the ACM.* 2002. To appear.

[2] Issarny, V. and Banâtre J-P. Architecture-based Exception Handling. *Proc. of HICSS'34.* 2001.

[3] Tartanoglu, F., Issarny, V., Levy, N. and Romanovsky, A., Dependability in the Web Service Architecture. *Proc. of WADS.* 2002.