THE ENGINEERING OF SOFTWARE
ERCIM News No.32 - January 1998

Architecture Support for the Testing of Complex Software Systems

by Antonia Bertolino and Paola Inverardi

We describe a new research project aimed at investigating methods and criteria for architecture-based software testing. The main partners in this collaboration are IEI-CNR, with several years of research experience in software testing, and the University of L'Aquila, well established in the field of software architectures.

The engineering of modern software systems is very hard. A principled use of Software Architecture (SA) in the design stage, when most crucial choices are made, helps to overcome system complexity, and hence can be a determining factor in development success. At a high level of abstraction, a Software Architecture provides an explicit and formal description of both the static and the dynamic properties of a system, ie the components and connectors. Our interest in the field of Software Architecture is in promoting the use of rigorous, formal descriptions of Software Architectures during development and, in particular, in order to improve analysis and testing.

The nature of research in Software Architectures makes it necessary, if not essential, to experiment on real-world systems. The scale of the applications considered is in fact one of the key aspects of the problem: the technical issues that arise when testing at the architectural level cannot be studied using simplistic, toy programs. Our project will thus involve several companies interested in improving the cost-effectiveness of their testing processes. These companies will act mainly as case study providers, but will also help us to keep the focus on substantial (and not purely academic) questions and will check the practicality of proposed solutions.

One case study is being provided by Hewlett Packard Italiana S.p.A., and consists of a network scanner. We are investigating methods for deriving an integration test plan, based on a formal description of the SA of the scanner. The main challenges of this case study are to guarantee portability over a multitude of platforms, and to meet the strict deadlines imposed by market competition. A second case study is provided by the Parco Scientifico e Tecnologico d'Abruzzo, a regional consortium of public and private research institutions and manufacturing industries. The system under development here is a Teleservice and Remote Medical Care System, which poses several safety and reliability constraints. We plan to involve other industries in the future.

Testing is one of the most expensive activities in software development and is mandatory to improve and evaluate the dependability of complex software systems. Traditionally, the testing of such systems proceeds in a series of successive phases, known as unit, integration and system tests. This step-wise approach on one hand allows testers to master the inherent complexity of the task, by subdividing the test object into smaller, more manageable pieces, and, on the other hand, enforces a disciplined procedure. However, we have identified several areas for improvement:

- It is often stated that the goal of the integration test is to check the compatibility between component interfaces. But static compatibility between interfaces is only one dimension of the problem: the design of software components naturally embodies assumptions about both the structure and the behaviour of the contexts in which they are to be integrated. Conflicts between components can thus be of a static nature, as well as dynamic. Conventional integration testing stresses structure over behaviour and is not well suited to uncovering dynamic conflicts.
- Automation is difficult and expensive, because the selection of test cases for the subsystems, and the sequencing order of the selected tests, depends on the functionality decomposition of the adopted system, which is not formalised. A new integration test strategy has thus to be re-invented ad hoc for each new system.
- When existing components are re-used in the integration of the system, as is often the case for large, complex systems, the test results obtained from the integration testing of the same components within different systems are not re-usable. The reason is that the role of components within the system, and the extent to which they are being tested, are not formally expressed. Given the high cost of test selection and execution, a capability to recover earlier test efforts would be highly desirable.

Our research activity is directed to addressing these problems by using the SA of the system as a primary reference. SA has recently emerged as an explicit field of study for software engineering researchers and practitioners. SA provides a disciplined basis for the engineering of large systems constructed out of many interacting components. In particular, since an SA provides an explicit and formal description of both the static and the dynamic properties of a system, it provides an important, if not essential, support towards making integration testing a more systematic and automatizable process. Not only do we expect to find suitable and workable solutions to the problems described above, but we also envisage several side benefits. For instance, when the design of an SA is based on the adoption of a specific architectural style, which constrains the components and the connectors to a restricted list, the testing itself can be tailored to the adopted style. We thus intend to investigate interesting architecture test patterns (in strict relationship with existing design patterns). Moreover, an SA typically specifies information about the extra-functional properties of components and connectors as well. Therefore, we also intend to investigate how to exploit the architecture specification to test for extra-functional requirements.

Please contact:

Antonia Bertolino - IEI-CNR
Tel: +39 50 593478
E-mail: bertolino@iei.pi.cnr.it

Paola Inverardi - Universita' di L'Aquila
Tel: +39 862 433127
E-mail: inverard@univaq.it

return to the contents page