Adaptable system/software architectures

Lawrence Chung a,*, Nary Subramanian b

a Department of Computer Science, The University of Texas at Dallas, Richardson, TX, USA
b Department of Computer Engineering, Hofstra University, Hempstead, NY, USA

Adaptability is emerging as an important type of non-functional requirement (NFR) for just about any system, including information systems, embedded systems, e-business systems, and the like. Briefly stated, adaptability is the ability of a system to accommodate changes in its environment. Since all systems work in an environment, an adaptable system should adapt to changes in its environment in order to succeed or even to survive. Often times the notion of adaptability seems to be quite broad, with many other closely-related types of NFRs such as flexibility, evolvability, transformability, reusability, and extensibility. This broad nature makes adaptability not only interesting but also critical in practice and challenging as a research topic.

Now at what point during the system/software development does one consider adaptability? It has been widely recognized that the earlier during the software development these NFRs are taken into account, the easier and the more cost-effective it becomes for the final system/software to satisfy these NFRs. The first step in the design of a system/software is the development of the underlying structure of the system/software, namely, the architecture. But how does one go about developing an adaptable system/software architecture?

The answer to this question was the focus of the Workshop on Adaptable System/Software Architectures (WASA) held in Las Vegas on June 25, 2003, the second in the series with the previous one held as a session of the International Conference on Software Engineering Research and Practice (SERP), Las Vegas, June 2002. The papers presented at WASA illustrated the problems and solutions for developing adaptable software architectures for different domains. In this special issue we have selected some of the more interesting papers among those presented at WASA. The selected papers exemplify the technical hurdles that need to be overcome to engineer adaptability in the respective domains. They also highlight the importance of the adaptability NFR in the various domains.

The papers for WASA came from six countries in three continents reflecting serious interests from both academia and industry. The first paper entitled “Mapping Evolving Business Rules to Software Design” by Wan M.N. Wan-Kadir and Pericles Loucopoulos, studies the relationship between evolution of business rules and evolution of software. The authors develop a link model that helps establish traceability between business requirements and software design. XML DTD is used to specify the metamodel of the UML design.

The second paper entitled “Assessing Systems Adaptability to a Product Family” by Mika Korhonen and Tommi Mikkonen, assesses the extension of an architecture for an industrial system into a product family—the problems encountered with this extension are assessed.

* Corresponding author.
E-mail address: chung@utdallas.edu (L. Chung).
and metrics have been used to assess the architecture.

The third paper entitled “Measures for Mobile Users: an Architecture” by Alberto Sillitti, Andrea Janes, Giancarlo Succi, and Tullio Vernazza, discusses the architecture of a tool called PROM that helps to automate collection of software metrics such as code length and reusability, as well as software process data such as time spent in each activity, number of changes per class, etc.

The fourth paper entitled “A Software System Evolutionary and Adaptive Framework: Application to Agent Based Systems” by Patricia Park-Padera-Rodriguez, Juan Jesus Torres-Carbonell, Ma Jose Rodriguez-Fortiz, Nuria Medina-Medina, and Fernando Molina-Ortiz, proposes a framework for the evolution of agent-based systems. The paper develops formalization for such systems using operators and detecting invariants.

The fifth paper entitled “A Model of Runtime Transformation for Distributed Systems Based on Directed Acyclic Graph Model” by Murakami Masaki, presents a formal model for runtime program transformation for distributed systems. The model uses linear logic and acyclic graphs to define program transformation mathematically.

The sixth paper entitled “Generative and Incremental Implementation for a Scripting Interface” by Vespe Savikko, describes the architecture of a tool that helps provide a flexible scripting support interface to existing systems. A real-life study implementing the architecture is also discussed.

We believe that these papers give a glimpse into adaptable problems in different domains and some techniques to develop adaptable architectures in these domains. We hope that the reader is able to get an insight into the important NFR of adaptability and its manifestation in different domains. We thank Rick Kazman, Tommi Mikkonen, Yeong-Tae Song, and Kendra Cooper for their prompt and thorough reviewing of the papers. We also thank Dr. Kuchcinski of Journal of Systems Architecture for agreeing to publish the special issue of JSA on WASA.