



3rd International Conference on the Quality of Software Architectures

"Software Architecture, Components, and Applications"

July 12-13, 2007,
Tufts University, Medford (Boston area), Massachusetts, USA
<http://qosa.ipd.uka.de>
qosa@ipd.uka.de

in conjunction with CBSE 2007 and ROSATEA 2007

– Accepted contributions will be published as Springer LNCS –



Motivation

Today, a systems's software architecture cannot be seen simply as a means to an end, the end being the implemented system. Although the ultimate measure of the quality of the software architecture lies in the implemented system, in how well it satisfies the requirements and constraints of the project and whether it can be maintained and evolved successfully, the quality of a system's software architecture is one of the critical factors in its overall system quality - encompassing both functional and extrafunctional properties. In order to treat design as an engineering discipline rather than an art, we need the ability to address the quality of the software architecture directly, not simply as it is reflected in the implemented system.

This is a specific goal of QoSA - to deal with software architecture in general and simultaneously focus on its quality characteristics by addressing the problems of:

- designing software architectures of good quality,
- defining, measuring, evaluating architecture quality, and
- managing architecture quality, tying it upstream to requirements and downstream to implementation, and preserving architecture quality throughout the lifetime of the system.

Cross-cutting these problems is the question of the nature of software architecture. Software architecture organizes a system, partitioning it into elements and defining relationships among the elements. For this we often use multiple views, each with a different organizing principle.

But software architecture must also support properties that are emergent and cannot be ascribed to particular elements. For this we often use the language of quality attributes. Quality attributes cover both internal properties, exhibited only in the development process (e.g. maintainability, portability, testability, etc.), and external properties, exhibited in the executing system (e.g. performance, resource consumption, availability, etc.). Quality attributes cover properties that are emergent, that have a pervasive impact, that are difficult to reverse, and that interact, thereby precluding or constraining other properties. Thus, QoSA also aims to investigate quality attributes in the context of the problems of the design, evaluation, and management of software architecture.

Topics

Topics of interest include, but are not limited to:

Architecture Design and Implementation:

- design decisions and their influence on the quality of software architecture
- organizational issues and processes that influence architecture quality
- architectural patterns and their quality impacts
- architectural standards and reference architectures
- integration of COTS components
- relationship between quality attributes and architectural design properties

Component Design and Implementation:

- how to design high-quality components that enable the construction of well-architected systems meeting functional and extrafunctional requirements
- specification and documentation of components and their quality attributes
- compositional reasoning and methods to predict architecture properties on the basis of component properties
- interface standards and standardization efforts
- component development methodology and development process
- component engineering and product-lines
- certification of components
- component markets and repositories
- services vs. components / service-oriented architecture vs. "classical" component-based architecture

Architecture Evaluation:

- lessons learned and empirical validation of theories and frameworks on architecture quality
- empirical validation of testing, prototyping, simulation for assessing architecture quality
- models and specification techniques to evaluate the quality attributes of software architectures
- processes for evaluating architecture quality
- evaluation of COTS components

Architecture Management:

- coordination of business architecture, business processes, and software architecture
- documentation of software architecture, including design rationale
- assessment and enforcement of architectural conformance
- traceability of software architecture to requirements and implementation
- assessment of COTS components
- integration of heterogeneous software architectures
- architecture evolution and architecture governance

Proceedings

QoSA welcomes both long and short papers. Long papers are up to 18 pages LNCS style, and can describe both research contributions and experience reports. Short papers are up to 8 pages LNCS style, and can describe experience, ongoing work, and new ideas.

As in the last years, accepted contributions will be published in a volume of the **Springer Lecture Notes in Computer Science** series. All Papers must be written in English and are to be prepared according to Springer's LNCS style (guidelines are available at: <http://www.springer.de/comp/lncs/authors.html>)

Important Dates

Papers due:	March 16, 2007 (extended)
Notification of acceptance:	April 20, 2007
Camera-ready versions due:	May 14, 2007
QoSA conference:	July 12-13, 2007

Venue

QoSA 2007 will be located in the United States and hosted by Tufts University in the Boston (Massachusetts) area. It will run jointly with CBSE 2007 and ROSATEA 2007 as the Federated Events on Component-Based Software Engineering and Software Architecture.

Program Committee Chairs

Sven Overhage,	Augsburg University, GER
Clemens Szyperski,	Microsoft, USA

Steering Committee

Ralf Reussner,	University of Karlsruhe, GER
Judith Stafford,	Tufts University, USA
Christine Hofmeister,	Lehigh University, USA
Steffen Becker,	University of Karlsruhe, GER

Program Committee

Antonia Albani,	Delft University of Technology, NL
Colin Atkinson,	University of Mannheim, GER
Len Bass,	Software Engineering Institute, USA
Don Batory,	University of Texas at Austin, USA
Jan Bosch,	Nokia Research Center, FIN
Alexander Brändle,	Microsoft Research, UK
Michel Chaudron,	Technische Universiteit Eindhoven, NL
Ivica Crnkovic,	Mälardalen University, SWE
Ian Gorton,	Pacific North West National Laboratory, USA
Hassan Gomaa,	George Mason University, USA
Volker Gruhn,	University of Leipzig, GER
Wilhelm Hasselbring,	University of Oldenburg / OFFIS, GER
Jean-Marc Jezequel,	University of Rennes / INRIA, FR
Philippe Kruchten,	University of British Columbia, CAN
Patricia Lago,	Vrije Universiteit, NL
Nicole Levy,	University of Versailles, FR
Tomi Mannisto,	Helsinki University of Technology, FIN
Nenad Medvidovic,	University of Southern California, USA
Raffaella Mirandola,	Politecnico di Milano, IT
Robert Nord,	Software Engineering Institute, USA
Frantisek Plasil,	Charles University, CZ
Iman Poernomo,	King's College, UK
Sasikumar Punnekkat,	Mälardalen University, SWE
Andreas Rausch,	University of Kaiserslautern, GER
Matthias Riebisch,	Technical University of Ilmenau, GER
Bernhard Rumpe,	University of Technology Braunschweig, GER
Christian Salzmänn,	BMW, GER
Jean-Guy Schneider,	Swinburne University, AUS
Johannes Siedersleben,	T-Systems, GER
Michael Stal,	Siemens, GER
Hans van Vliet,	Vrije Universiteit, NL
Kurt Wallnau,	Software Engineering Institute, USA
Wolfgang Weck,	Independent Software Architect, CH