

Special Session on Formal and Quantitative Approaches for Non-Functional Requirements Modeling and Assessment in Software Engineering

M. Kassab, O. Ormandjieva
Department of Software Engineering and
Computer Science
Concordia University
Montreal, Canada
{moh_kass , ormandj}@cs.concordia.ca

M. Daneva
Department of Information Systems
University of Twente
Enschede, the Netherlands
m.daneva@utwente.nl

1. Scope of the Session

Once a software system has been deployed, it is typically straightforward to observe whether or not a certain Functional Requirement (FR) has been met, as the areas of success or failure in their context can be rigidly defined. However, the same is not true for Non-Functional Requirements (NFRs) as these can refer to concepts that can be interdependent and difficult to measure.

The problem of lacking any early NFR integration within the specified system is likely to cause an increase in the effort and maintenance overhead. The importance of software compliance with the imposed NFRs requires management of their scope, which brings up the importance of clearly defining, tracing and effort estimating the complex and frequently ill-defined NFRs and their interrelations in increasingly complex large-scale software system.

This session aims to provide a forum for both practitioners and researchers to discuss the most recent advances in theory and practice addressing the modeling and assessment of NFRs at the early project stages. The session topics include but are not limited to the following topics:

- **NFRs Conceptualization:** In general, and because of their diverse nature, NFRs have been (at best) specified in loose, fuzzy terms that are open to wide ranging and subjective interpretation. As such, they provide little guidance to architects and engineers as they

make the already tough trade-offs necessary to meet schedule pressures and functionality goals. For instance, most software engineering approaches and industrial practices specify NFRs separately from FRs of a system. This is mainly because the early integration of NFRs is difficult to achieve and usually accomplished at the later phases of the software development process. However, since the integration is not supported from the requirements phase to the implementation phase, some of the software engineering principles such as abstraction, localization, modularization, uniformity and reusability, can be compromised. Furthermore, the resulting system is more difficult to maintain and evolve.

Instead, NFRs need to be made precise and clear right from the requirements phase. But in order to be able to specify the NFRs in precise terms, there must be a general understanding to what the term NFR stands for, and what are the relations that the NFR may be exposed to during the lifecycle of the project. In fact, although the term “non-functional requirement” has been in use for more than 20 years and a few approaches to conceptualizing NFRs have been proposed re, there is still no consensus in the requirements engineering community about what NFRs are, what the relations are that an individual NFR may participate in, and how reasoning about NFRs and functional requirements can take place in an integrated fashion.

- **NFRs Traceability:** Recent publications in requirements engineering indicate a multifaceted gap between the requirements and the developed solution in a project. Traditional software

development approaches do not address this gap. For example, those architectural design methods, that link architecture to requirements, make architecture a central concern. These methods use requirements as input only or as a standard for evaluation. They, however, disregard the ability of software architecture to inform the requirements engineering processes and also the need to trace architecture design choices back to requirements. More often than not, existing approaches fail to convey change, rationale, options, and organizational implications of requirements or of solution designs. The gap between requirements engineering and solution design seems to be essentially a problem of traceability.

Traceability has so far been tackled mainly qualitatively, identifying related elements in requirements engineering and solution to visualize those elements of solution impacted by changing requirements and vice versa.

Tracing NFRs from requirements engineering to solution design poses further challenges as these requirements tend to scatter among multiple modules when they are mapped from the requirements domain to the solution space. Another challenge arises as the existing approaches to model NFRs lack an adequate specification of the semantics of NFRs, which leads to inconsistent interpretational uses of these requirements.

- Effort Estimation of Building NFRs: NFRs are very challenging when estimating the effort and the time it would take to implement them. This is mainly because of the unique nature of these requirements: NFRs are subjective, relative, interacting and crosscutting. However, it is crucial to be able to make decisions about the scope of software by given resources and budget based on a proper estimation of building both FRs and NFRs.

2 Forms of Participation

Research papers are either describing a working experience at the intersection of architecture design and engineering the NFRs in a project or the results of a research effort related to the session topics. The research papers should clearly describe the problem tackled, the relevant

state of the art, the position or solution suggested and the potential benefits of the contribution.

Papers will be submitted at the SERA2010 website in PDF format following the SERA2010 template.

The papers are due on: February 1st, 2010. The authors will receive acceptance notifications on March 1st, 2010. The camera ready papers & registration deadline is March 15th, 2010.

The selection procedure includes a review of each paper by at least three PC members. Experience papers will be accepted based on the relevance of the problems they tackle. Research papers will be accepted based on the relevance of the research questions to practice and the soundness of the research method deployed.

3 ORGANIZERS' BIO-SKETCH

Dr. Mohamad Kassab has a PhD in Computer Science from Concordia University in Montreal. His main research interest includes a formal and quantitative approaches towards the integration of NFRs throughout the development process of a software product. He authored the book: "Non-Functional Requirements: Modeling and Assessment". Dr. Kassab has more than 10 years of professional work experience in the IT and software industry. Currently he is working in quality assurance at Nokia-Montreal.

Dr. Olga Ormandjieva is an associate professor with the Computer Science and Software Engineering department, Concordia University. The main area of research of Dr. Ormandjieva is Measurement in Software and Service Engineering (functional size, reliability, performance, Quality of Services, Quality of Experience) and its extension to the development of formal methods for modelling and monitoring complex systems with Category Theory. Her research interests include automation of the functional size measurement and project effort estimation from business and software requirements text.

Dr. Maya Daneva is an assistant professor with the Information Systems group at the University of Twente, where she leads a research program on ERP requirements engineering and functional size and cost estimation. She serves as a research liaison to the companies in the Netherland's IT consulting service industry. Prior to this, Maya worked 9 years on requirements-based size and effort estimation models for large projects in the telecommunication sector as a business process analyst in the Architecture Group of TELUS Corporation, Toronto, where she consulted on ERP requirement engineering processes and requirements-based size and effort estimation for ERP. Maya authored more than 60 research and experience papers.