ARAMIS 2008
the First Int. Workshop on
Automated engineeRing of Autonomic and run-tiMe evolvIng Systems

Mauro Caporuscio, INRIA Paris-Rocquencourt, France, mauro.caporuscio@inria.fr
Antinisca Di Marco, University of L'Aquila, Italy, adimarco@di.univaq.it
Leonardo Mariani, University of Milano Bicocca, Italy, mariani@disco.unimib.it
Henry Muccini, University of L'Aquila, Italy, muccini@di.univaq.it
Andrea Polini, University of Camerino, Italy, andrea.polini@unicam.it
Onn Shehory, IBM Haifa Research Lab, Haifa University Campus, ISRAEL, onn@il.ibm.com

www.selab.isti.cnr.it/Aramis2008/

Abstract

An increasingly important requirement for modern software-intensive systems is the ability of evolving at run-time, to address the need to dynamically add/remove features, as well as to protect the system from incoming attacks or failures. Due to the new requirement of dynamicity, portions of the system can be added and removed at run-time so giving rise to dynamic, evolving, and unpredictable configurations, implemented via middleware-based technologies, plugin-based infrastructures, or service oriented paradigms.

The ARAMIS workshop wants to provide a forum for analyzing the new challenges and issues related to evolving at run-time systems. In particular, the main objective is to analyze such topics from three different, but correlated, perspectives: verification and validation of evolving at run-time systems, automatic context management, and self-healing solutions for engineering such systems.

1. Workshop Theme and Goals

Modern software-intensive systems are often large, continuously running applications, deployed on complex and geographically distributed environments. These systems are characterized by a high rate of changes and unexpected situations that must be dynamically and automatically handled. Changes and unexpected situations can have many sources: the need to dynamically add/remove features to accommodate new user requirements, the necessity to adapt to (physical and logical) context changes, the necessity to protect from failures and to automatically repair faults. Such needs introduce new challenges that cannot be exclusively addressed with traditional solutions; rather, they require the development of highly automated techniques. That is, automated techniques for managing, verifying and validating systems evolving at run-time, automated techniques for handling, overcoming and fixing system failures, and automated techniques for adaptation to context changes.

The evolutionary nature of modern software-intensive applications makes infeasible a standard approach to (functional and extra-functional) verification and validation. The focus must move from validating and removing faults from design time to validating and healing evolving systems at run-time. While in static systems, quality assurance tasks like verification, validation, testing, debugging and fixing can be done once and for all before deployment, for systems changing at run-time they become a perpetual activity to be performed during system execution. Changes not only take place in applications, but also in the environment. Thus, suitable techniques for automatically capturing context changes and reasoning on context information to properly adapt to changing environments must be embedded into software systems. The execution of such activities in the field imposes real-time requirements that must be satisfied by all the analysis techniques working at run-time.

The ARAMIS workshop has provided a forum for scientists and engineers in academia and industry to present and discuss their latest research. ARAMIS focuses on issues, challenges, and future perspectives of techniques for automated verification and validation of evolving systems, for automatic context management and for the engineering of self-healing solutions.

Topics of interest include:

Verification and validation of run-time evolving sys-

978-1-4244-2188-6/08/$25.00 ©2008 IEEE

tems: this macro-area of interest includes many aspects of interest. *The impact of run-time Evolution on v&v* that covers understanding of run-time evolution requirements and characteristics, ways to model run-time evolving systems, and comparative analysis between v&v of traditional system versus v&v of run-time evolving systems. *Approaches* for v&v of run-time evolving systems that deal with different techniques for verifying and validating systems changing at run-time, with a particular emphasis on monitoring, online testing, and fault tolerance. Verification and validation in specific *Application Domains* that includes v&v of Service Oriented Architectures, and v&v of Self-* systems.

Automatic Context Management: this macro-area of interest includes several aspects. *Context Definition* that covers context metamodeling and ontologies, models and languages to describe contexts, tools supporting the context definition and description. *Context Management* that includes monitoring of the context characteristics and detection of context changes. Run-time *Mechanisms* to adapt software systems to context changes and to reason on context information in order to support the software system adaptation. *MDE techniques* for the generation of code and QoS validation models to support the development of context-aware software systems. Context-based run-time *QoS models* that can guide the adaptation of the software system to the context changes.

Engineering of self-healing solutions: this macro-area of interest includes several aspects. *Self-Healing Solutions for Functional and Non-Functional Faults* that covers techniques for automatically detecting failures, identifying problem causes, and repairing (functional and non-functional) faults. *Architectures for Self-Healing Solutions* that covers the definition of architectures and architectural styles that can be used to design self-healing techniques. *Design for Self-Healing* that covers the design of systems to be augmented with self-healing capabilities. *Model-Based Approaches to Self-Healing* that covers the definition of models and model-based analysis techniques that can be used as part of self-healing solutions. *Automated Debugging Techniques* that covers testing-time solutions for automatically suggesting fixes to faults that have been revealed.

2. Workshop Format and Summary

Important goal of the workshop is to foster the discussion, in order to identify relevant research lines to pursue, and to encourage collaborations among the workshop participants. With this goal in mind, the workshop has been organized as a one-day event, consisting of a series of sessions, each devoted to the presentation and discussion of papers belonging to a common domain. Three types of sessions have been planned: *technical papers* presentation sessions, a *research project* panel session, and a *discussion session*.

During the technical papers sessions, technical papers are presented and discussed. Paper presentations will last fifteen minutes plus five minutes for questions. At the end of each technical paper session a round table will start, discussing interesting points raised by the presentations.

In order to brighten up and to widen the discussion we planned a *research project panel session* where several research projects, related to ARAMIS topics, have been invited to participate. The project coordinators have been also invited to submit a short project paper that points out: the main objectives of the project and how it relates to ARAMIS topics, the main challenges and issues encountered during the project progress, the final expectations, and directions for further work on the topic. As any other papers, project papers have been peer reviewed in order to be published in IEEE DL. PLASTIC (http://www.ist-plastic.org/), SHADOWS (https://sysrun.haifa.il.ibm.com/shadows/index.html), Music (http://www.ist-music.eu/) and Q-ImPrESS (http://www.q-impress.eu/Q-ImPrESS/CMS/index_html) are those projects participating to ARAMIS.

A *discussion session* has been also planned for discussing the overall results coming from the workshop and plan for future editions.

3. Program Committee

Nelly Bencomo (*Lancaster University* - UK), Antonia Bertolino (*ISTI/CNR* - Italy), Flavio Corradini (*University of Camerino* - Italy), David Garlan (*Carnegie Mellon University* - USA), Nikolaos Georgantas (*INRIA P.R.* - France), Holger Giese (*HPI at University of Postdam* - Germany), Ian Gorton (*Pac. Northwest Nat. Lab.* - USA), Paul Grace (*University of Lancaster* - UK), Marta Kwiatkowska (*Trinity College Oxford* - UK), Bohuslav Krena (*University of Brno* - Czech Republic), Jeff Magee (*Imperial College London* - UK), Tiziana Margaria (*University of Postdam* - Germany), Mauro Pezzè (*University of Milano Bicocca* - Italy), Iman Poernomo (*King's College London* - UK), Alexander Romanovsky (*University of Newcastle* - UK), Stefano Salsano (*University of Tor Vergata* - Italy), Shmuel Ur (*IBM Haifa Res. Labs* - Israel), Dániel Varró (*University of Budapest* - Hungary), Michel Wermelinger (*The Open University* - UK), Andrea Zisman (*City University London* - UK)

Acknowledgements

We would like to thank the authors of accepted papers and also all submitters, and all PC members, who actively contributed to the ARAMIS call and reviewed with accuracy the papers under a tight schedule.