Lecture Notes in Computer Science 5835

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Rogério de Lemos Jean-Charles Fabre
Cristina Gacek Fabio Gadducci
Maurice ter Beek (Eds.)

Architecting Dependable Systems VI

Springer

Volume Editors

Rogério de Lemos
University of Kent, Computing Laboratory
Canterbury, Kent CT2 7NF, UK
E-mail: r.delemos@kent.ac.uk

Jean-Charles Fabre
LAAS-CNRS
7, avenue du Colonel Roche, 31077 Toulouse Cedex 4, France
E-mail: jean-charles.fabre@laas.fr

Cristina Gacek
Newcastle University, School of Computing Science
Newcastle upon Tyne, NE1 7RU, UK
E-mail: cristina.gacek@ncl.ac.uk

Fabio Gadducci
Università di Pisa, Dipartimento di Informatica
Largo Pontecorvo 3c, 56127 Pisa, Italy
E-mail: gadducci@di.unipi.it

Maurice ter Beek
Istituto di Scienza e Tecnologie dell'Informazione (ISTI-CNR)
Area della Ricerca di Pisa, Via G. Moruzzi 1, 56124 Pisa, Italy
E-mail: maurice.terbeek@isti.cnr.it

Library of Congress Control Number: 2009937881

CR Subject Classification (1998): D.2, D.4, B.8, D.1.3, F.1.2, K.6.5, D.4.6, E.3
LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-642-10247-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-10247-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12780082 06/3180 5 4 3 2 1 0

Foreword

The challenges that arise from building and running large enterprise applications are among the most daunting and underappreciated in computer science today. Such applications are invariably large, with millions of lines of code. They are often executed on multiple machines that may be located in different geographic areas and connected by networks of various speeds and capabilities, leading to issues related to distribution, concurrency, and networking. Given the importance of these applications to the financial health and stability of the companies involved, the requirements related to dependability—reliability, availability, timeliness, security—are also often strict. Finally, applications from one enterprise increasingly need to interoperate programmatically in a seamless fashion with applications from other companies to support business-to-business (B2B) transactions, whether with suppliers, customers, or peers. In short, enterprise applications are difficult to construct, operate, and maintain, yet are a critical part of the world's economic infrastructure.

This volume brings together the efforts of researchers from the dependability and software architecture communities to address issues important for solving the dependability challenges of enterprise applications. The first group of papers deals with dependability in the context of the service-oriented architecture (SOA) structuring paradigm. With an SOA, software functionality in a distributed system is structured as collections of interacting services, possibly operated by different companies or otherwise in distinct administrative domains. The services include both infrastructure services, such as directory services, monitoring, and resource allocation services, as well as application services that implement some application-specific functions. A given service is usually represented by one or more published interfaces, which allow other services to find and access it dynamically at runtime. In their pure form, SOAs offer fundamental characteristics that can simplify the construction and operation of enterprise applications, including support for dynamic operation, long-term software evolution, extensibility, and composibility.

While SOAs are potentially a useful way to address some of the issues that surround enterprise applications, they are not a panacea and in many ways merely reorient the inherent problems in this domain into a new perspective. This observation does not diminish their value, however, since it is often the viewing of existing challenges through a new lens that leads to the creation of new abstractions, techniques, and methodologies. This is especially true for dependability attributes, which are notoriously difficult to ensure in large heterogeneous distributed systems like those for which SOAs are intended. In this context, SOAs and their related technologies are important and timely topics that need to be addressed by the research community, and it is indeed fortunate to have collected here a number of papers that do just that.

The second group of papers addresses issues related to the evaluation of systems with critical dependability attributes. Evaluation is, of course, a key step for any system, but it is especially important for enterprise applications given their economic importance and strict dependability requirements. Bringing to bear tools and techniques from both the dependability and architecture communities is a good starting point for developing approaches that can eventually scale to the size and complexity of collections of interacting enterprise applications.

The final group of papers focuses on architecting security. The importance of this attribute for enterprise applications cannot be underestimated, and it is arguably the foundational element for any type of enterprise-oriented system. Without guarantees related to privacy, authentication, integrity and similar properties, it is impossible to interact with customers, suppliers, or peers in any kind of rational and safe way. Again, the application of architectural principles from software engineering coupled with dependability techniques provide a compelling vantage point from which to make progress.

As is clear from the above, perhaps the most unique aspect of this collection of papers is how it represents the best ideas from research in both software architectures and dependability. It is refreshing to see two traditionally separate communities coming together to address problems not only of common interest, but also of critical importance to society. The dependability issues associated with enterprise applications are deep and challenging, and the papers in this collection are indeed a welcome addition to the literature in this area.

August 2009

Rick Schlichting
AT&T Labs – Research

Preface

This is the sixth book in a series on Architecting Dependable Systems. This series started seven years ago, and brings together issues related to software architectures and the dependability and security of systems. This book includes expanded and peer-reviewed papers based on the selected contributions to two workshops, and a number of invited papers written by recognized experts in the area. The two workshops were: the Workshop on Architecting Dependable Systems (WADS) organized at the 2008 International Conference on Dependable Systems and Networks (DSN 2008), and the Third International Workshop on Views On Designing Complex Architectures (VODCA 2008).

Identification of the system structure (i.e., architecture) early in its development process makes it easier for the developers to make crucial decisions about system properties and to justify them before moving to the design or implementation stages. Moreover, the architectural level views support abstracting away from details of the system, thus facilitating the understanding of broader system concerns. One of the benefits of a well-structured system is the reduction of its overall complexity, which in turn leads to a more dependable and secure system. System dependability is defined as the reliance that can be justifiably placed on the service delivered by the system, while security can be defined as protecting the system and certain information it contains from unauthorized access and handling. Both have become essential aspects of computer systems as everyday life increasingly depends on software. It is therefore a matter of concern that dependability and security issues are usually left until too late in the process of system development.

Making decisions and reasoning about structure happen at different levels of abstraction throughout the software development cycle. Reasoning about dependability at the architectural level has recently been in the focus of researchers and practitioners because of the complexity of emerging applications. From the perspective of software engineering, traditionally striving to build software systems that are fault free, architectural consideration of dependability requires the acceptance of the fact that system models need to reflect that it is impossible to avoid or foresee all faults. This requires novel notations, methods and techniques providing the necessary support for reasoning about faults (including fault avoidance, fault tolerance, fault removal and fault forecasting) at the architectural level. Moreover, due to the inherent design trade-off between dependability and security attributes, security issues should also be taken into account at the architectural level.

This book comes as a result of bringing together research communities of software architectures, dependability and security, and addresses issues that are currently relevant to improving the state of the art in architecting dependable and secure systems. The book consists of three parts: Dependable Service-Oriented Architectures, Fault Tolerance and System Evaluation, and Architecting Security.

The first part entitled “Dependable Service-Oriented Architectures” includes five papers focusing on various aspects on how to design dependable service-oriented

systems. The first paper of this part, authored by R. Jimenez-Peris, M. Patiño-Martinez, B. Kemme, F. Perez-Sorrosal and D. Serrano, and entitled "A System of Architectural Patterns for Scalable, Consistent and Highly Available Multitier Service-Oriented Infrastructure" describes how, in the context of service-oriented architectures, replication can be performed across a multi-tier architecture in order to satisfy the high availability, consistency and/or scalability requirements. These architectural patterns can guide system architects and practitioners in evaluating and selecting the appropriate architectural choices in order to replicate multi-tier software infrastructures.

V. Cardellini, E. Casalicchio, V. Grassi, F. Lo Presti and R. Mirandola contribute to the book with the paper "Towards Self-Adaptation for Dependable Service Oriented Systems." This paper proposes a model-based approach to the realization of self-adaptable systems adopting the service-oriented architecture (SOA) paradigm, aimed at the fulfilment of dependability requirements. It introduces a methodology driving the system adaptation highlighting the architectural issues related to its implementation. This is achieved by means of the presentation of a possible architecture for this type of systems, which can be seen as an instantiation for the SOA environment of the general architectural framework for self-adapting systems proposed within the autonomic computing initiative. Given this architecture, the focus is then on determining suitable adaptation actions in response to detected environmental changes.

M. P. Machulak, S. E. Parkin, and A. van Moorsel contribute to the book with the paper "Architecting Dependable Access Control Systems for Multi-Domain Computing Environments." This paper reviews the state of the art in requirements analysis for authorization mechanisms in highly distributed multi-domain computing environments, focussing in particular on environments that are built on SOAs that interact through Web Services. This analysis provides a comprehensive insight into both existing and future authorization mechanisms.

The fourth paper, written by S. Bistarelli and F. Santini, is entitled "Soft Constraints for Dependable Service-Oriented Architectures." The paper aims at extending quality of service measures of SOAs with aspects of dependability. The challenge is to represent (as well as evaluate and improve) dependability as an architectural feature, rather than an implementation property. The key idea of the paper is to use the soft constraint framework (extending the classical constraint notions) in order to be able to manage SOAs in a declarative fashion by considering together the requirements/interfaces of each service and their dependability estimation.

The final paper of this part, entitled "Robustness Validation in Service-Oriented Architectures" and written by N. Laranjeiro, M. Vieira and H. Madeira, addresses the problem of robustness validation in SOA environments. It builds on previous work to provide a generic approach for the definition of robustness benchmarks for service based environments. The approach proposed is based on a set of robustness tests that is used to discover both programming and design errors. The paper is grounded by an illustration using two concrete examples, one focusing on Web services and the other targeting Java Message Service (JMS) middleware.

The second part of this book is entitled "Fault Tolerance and System Evaluation" and contains four papers. The first paper, entitled "A Self-Repair Architecture for Cluster Systems," is written by F. Boyer, N. De Palma, O. Gruber, S. Sicard and J.-B.

Stefani. This paper presents a framework for the construction of self-repairable cluster systems. Self-repair is achieved in the JADE framework through a combination of component-based design, reflection and active replication of the management subsystem. This architecture-based management system is able to tolerate partial failures and allow failed subsystems to be repaired and reinserted without requiring a global shutdown.

The paper "Handling Software Faults with Redundancy" by A. Carzaniga, A. Gorla and M. Pezzè present a survey of several techniques for handling software faults that were developed in the areas of fault tolerance and autonomic computing. This paper considers the impact of redundancy on the software architecture, and proposes a taxonomy centered on the nature and use of redundancy in software systems.

G. Lenzini, F. Martinelli, I. Matteucci, and S. Gnesi contribute to the book with the paper "A Uniform Approach to Security and Fault-Tolerance Analysis." The paper moves from the recognition that, while dependability analysis of distributed systems is dominated by fault-tolerance and security, these two disciplines evolved in parallel, cross-breeding yet developing tools and techniques separately. Thus, the authors illustrate how two security analysis techniques, related to partial model checking and to so-called generalized non-interference, can be applied to verify a family of fault-tolerance properties. Moreover, exploiting previous results concerning the framework of non interference analysis, some compositional analysis techniques are shown to be available.

The final paper of this part, entitled "A Comprehensive Exploration of Challenges in Architecture-Based Reliability Estimation" and written by I. Krka, G. Edwards, L. Cheung, L. Golubchik and N. Medvidovic, provides guidance for enhancing the existing architecture-based reliability estimation approaches and motivates the development of new techniques. In this direction, the authors have identified three core challenges in architecture-based reliability estimation: defining a failure model, obtaining reliability-related parameter estimates, and dealing with the scale and complexity of modern software. They have outlined each of these challenges, and described promising solutions to them.

Part three of the book is on "Architecting Security" and includes four papers focusing on security at the architectural level. The first paper, entitled "Weak Behavioral Equivalences for Verifying Secure and Performance-Aware Component-Based Systems" and written by A. Aldini and M. Bernardo, proposes a two-phase predictive methodology whose goal is to balance the trade-off between security and performance (QoS) in system design. The first phase uses the functional noninterference approach to security analysis, while the second phase uses nonfunctional performance analysis. The methodology is applied to the stochastic process-algebraic architectural description language AEmilia and it is illustrated through its application to a running example based on a multilevel security routing system.

The second paper is written by S. Andova, L. P.J. Groenewegen, J. H. S. Verschuren and E. P. de Vink, and entitled "Architecting Security with Paradigm." This paper describes a modelling suite for architecting the security protocols of software designs. The coordination language Paradigm is used to model the collaborating components, specifically taking into account the dynamic consistency between the architectural components. Subsequently a translation into process algebra allows model

checking with the state-of-the-art toolset mCRL2. Two case studies illustrate the approach.

G. Lenzini contributed to the book with the paper "Trust-Based and Context-Aware Authentication in a Software Architecture for Context and Proximity-Aware Services". This paper describes an existing software architecture for trust prediction in the setting of proximity-Aware services with unobtrusive and context-based authentication capabilities. A user's presence is predicted with a certain level of trust by combining information from a sensor network concerning the location of the user's personal devices. Each sensor is seen as a recommender providing an opinion, which are collected, interpreted and weighted by a context management framework. Experiments illustrate the reliability of the identification and authentication algorithm in a test-case scenario.

The final paper of this part is entitled "Compositional Verification of Architectural Refactorings" and authored by D. Bisztray, R. Heckel and H. Ehrig. Motivated by the adoption of model-driven development, the paper presents a methodology for model refactoring that allows for the preservation of chosen behavioral properties. More precisely, the authors propose a heuristics for verifying the transformations of UML architectural models based on a semantic mapping into CSP processes. Suitable conditions are provided, in order to guarantee the correctness of such a method, and the feasibility of the verification, by requiring that the mapping satisfy certain compositionality properties.

Architecting dependable systems is now a well-recognized area, attracting interest and contributions from many researchers. We are certain that this book will prove valuable for both developers designing complex applications and researchers building techniques supporting this. We are grateful to many people that made this book possible. Our thanks go to the authors of the contributions for their excellent work, the DSN 2008 WADS and VODCA 2008 participants for their active participation in the discussions. We would also like to thank Alfred Hofmann and his team from Springer for believing in the idea of a series of books on this important topic and for helping us to get it published. Last but not least, we greatly appreciate the efforts of our reviewers who have helped us in ensuring the high quality of the contributions. They are Alessandro Aldini, Suzana Andova, Paris Avgeriou, Roberto Baldoni, Stefano Bistarelli, David Chadwick, Cas Cremers, Erik de Vink, Elisabetta Di Nitto, Jurgen Doser, Stefania Gnesi, Vincenzo Grassi, Luuk Groenewegen, Lars Grunske, Robert S. Hanmer, Reiko Heckel, Matthias Hölzl, Ricardo Jimenez-Peri, Jan Jürjens, Gabriele Lenzini, Henrique Madeira, Fabio Martinelli, John D. McGregor, Neno Medvidovic, Veena B. Mendiratta, Raffaela Mirandola, Henry Muccini, Marta Patiño-Martinez, Marinella Petrocchi, Leila Ribeiro, Francesca Rossi, Aad van Moorsel, Marco Vieira and several anonymous reviewers.

August 2009

Rogério de Lemos
Jean-Charles Fabre
Cristina Gacek
Fabio Gadducci
Maurice ter Beek

Table of Contents

Part 1. Dependable Service-Oriented Architectures

Part 2. Fault Tolerance and System Evaluation

Part 3. Architecting Security