Lecture Notes in Computer Science 5135

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

Rogério de Lemos Felicita Di Giandomenico
Cristina Gacek Henry Muccini
Marlon Vieira (Eds.)

Architecting Dependable Systems V

Springer

Volume Editors

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

Felicita Di Giandomenico
ISTI-CNR, Area della Ricerca CNR
Via G. Moruzzi 1, 56124 Pisa, Italy
E-mail: felicita.digiandomenico@isti.cnr.it

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

Henry Muccini
Università dell'Aquila, Dipartimento di Informatica
Via Vetoio, 1, 67010 L'Aquila, Italy
E-mail: muccini@di.univaq.it

Marlon Vieira
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540, USA
E-mail: marlon.vieira@siemens.com

Library of Congress Control Number: 2008933382

CR Subject Classification (1998): D.2, D.4, B.8, E.1

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-85570-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-85570-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12446032 06/3180 5 4 3 2 1 0

Preface

This is the fifth book in a series on Architecting Dependable Systems we started six years ago that brings together issues related to software architectures and the dependability 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 2007 International Conference on Dependable Systems and Networks (DSN 2007), and the Third Workshop on the Role of Software Architecture for Testing and Analysis organized as part of a federated conference on Component-Based Software Engineering and Software Architecture (CompArch 2007).

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 system. System dependability is defined as the reliance that can be justifiably placed on the service delivered by the system. It has become an essential aspect of computer systems as everyday life increasingly depends on software. It is therefore a matter for concern that dependability 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.

This book comes as a result of bringing together research communities of software architectures and dependability, and addresses issues that are currently relevant to improving the state of the art in architecting dependable systems. The book consists of three parts: “Critical Infrastructures,” “Rigorous Design and Fault Tolerance,” and “Verification and Validation.”

The first part entitled “Critical Infrastructures” includes six papers focusing on various aspects of architecting critical infrastructures. The structuring of software systems at the architectural level is especially fundamental for the development of critical infrastructures. Nowadays, public health, economy, security and quality of life

heavily depend on the resiliency of a number of critical infrastructures, including energy, telecommunications, transportation, emergency services and many others. The technological advances and the necessity for improved efficiency resulted in increasingly automated and interlinked infrastructures, with consequences on increased vulnerabilities to accidental and human-made faults. Addressing the development of such systems with rigorous methodologies and evolutionary approaches at an architectural level has high potential to enhance dependability and resiliency in these critical sectors. In recognition of this emerging necessity, this book includes a specific part on architecting critical infrastructures. The six contributions grouped in this section focus on architecting critical infrastructures at both the system design level, where intrusion tolerant architectures, virtualization, improved middleware technologies and efficient communication infrastructures are addressed, and at the verification and validation level, where the problem of modeling and understanding interdependencies among interlinked critical infrastructures is tackled.

The first paper in this part, "The CRUTIAL Architecture for Critical Information Infrastructures," is by P. Verissimo, N. Neves, M. Correia, Y. Deswarte, A. Abou El Kalam, A. Bondavalli and A. Daidone. In this paper the authors highlight the susceptibility of critical information infrastructures to computer-borne attacks and faults. They discuss how to overcome these problems and propose a generic architecture as well as a set of techniques and algorithms aiming at achieving resilience of critical information infrastructures to faults and attacks in an automatic way.

The paper "A Middleware Improved Technology (MIT) to Mitigate Interdependencies Between Critical Infrastructures" by C. Balducelli, A. Di Pietro, L. Lavalle and G. Vicoli deals with new middleware technologies (MIT) to support co-ordination among different large complex critical infrastructures (LCCI). The objective is to mitigate interdependency effects so as to enhance the resilience and survivability of LCCIs. The features provided by the MIT technology, as well as the adopted reference architecture and an experimental environment for testing purposes, are overviewed. The paper is developed in the framework of the EU IRRIIS project, which aims at protecting critical infrastructures in the energy and telecommunication domains.

S. Chiaradonna, F. Di Giandomenico and P. Lollini contribute to the book with the paper "Evaluation of Critical Infrastructures: Challenges and Viable Approaches." This paper introduces critical infrastructures with a focus on the challenges for evaluation. Furthermore, it provides initial results from a Moebius modeling framework to evaluate failures in the ICT infrastructure of an electric power system. The experience gained by the authors in a European project is reported and discussed.

The fourth paper, written by A. Daidone, S. Chiaradonna, A. Bondavalli and P. Verissimo is entitled "Analysis of a Redundant Architecture for Critical Infrastructure Protection." In the CRUTIAL reference architecture each LAN is connected to the WAN through a special interconnection and filtering device. Replica rejuvenation strategy applied to these devices is based on both periodic (proactive) recoveries and on event-triggered (reactive) recoveries, seeking perpetual unattended correct operation. This paper analyzes the redundant architecture of these devices by evaluating how effective the trade-off between proactive and reactive recoveries is, identifying the relevant parameters of the architecture and finding the best parameter setup.

The fifth paper in this part, "A Robust Semantic Overlay Network for Microgrid Control Applications," is by G. Deconinck, T. Rigole, H. Beitollahi, R. Duan, B. Nauwelaers, E. Van Lil, J. Driesen, R. Belmans and G. Dondossola. In this paper the authors present Agora, which is a semantic overlay network that allows one to efficiently route queries in overlay networks. The routing is related to microgrid control, and the semantics is based on an XML description of the static and dynamic characteristics of the intelligent electronic devices. It is robust against changes, and provides graceful degradation in case of unrecovered failures.

The paper "Architecting Dependable and Secure Systems Using Virtualization" by B. Jansen, H. V. Ramasamy, M. Schunter and A. Tanner explores the emerging virtualzation approach, through which the real hardware system configuration is abstracted from, to enhance systems dependability and security. A practical realization of a subset of the proposed enhancements, namely, intrusion detection and protection, using the Xen open-source virtual machine monitor (VMM) is detailed. In such a context, the impact of virtualization on node reliability is quantified using combinatorial modeling. The results of these analyses constitute useful guidelines on design options to effective leveraging of virtualization to system dependability purposes. They also triggered further improvements by the authors on the VMM design.

The second part of this book is entitled "Rigorous Design and Fault Tolerance" and contains three papers.

The paper "Model-Based Approaches for Dependability in Ad-Hoc Mobile Networks and Services" by G. Pinter, Z. Micskei, A. Kovi, Z. Egel, I. Kocsis, G. Huszerl and A. Pataricza reports the authors' research activity to architect dependable, distributed systems through a model-driven design approach. The ad-hoc mobile networks context adopted by the EU Hidenets project is specifically addressed. Contributions include the construction of the UML model of the Hidenets platform, the construction of a metamodel of the applications running on the platform, the UML profile for the metamodel as well as the definition of a set of design patterns to support the implementation of applications built for the Hidenets platform using the defined profile.

In the paper "Design, Implementation and Deployment of State Machines Using a Generative Approach", G. N.C. Kirby, A. Dearle and S.J. Norcross present an approach to designing and implementing a distributed system as a family of related finite state machines, generated from a single abstract model. The state machine family formalizes the interactions between the components of the distributed system, allowing increased confidence in correctness. The feasibility of the proposed approach was demonstrated in the context of a Byzantine fault-tolerant commit protocol used in a distributed storage system.

The third paper in this part, "Handling Emergent Nondeterminism in Replicated Services," is written by J. Slember and P. Narasimhan. This paper presents Midas, an approach to identifying and addressing multiple sources of nondeterminism in a multi-service replicated distributed architecture. Midas involves a combination of compile-time dependency, concurrency and nondeterminism analyses, followed by the performance-sensitive compensation of nondeterminism at runtime.

Part three of the book is on "Verification and Validation" and includes five papers focusing on approaches to architecture level verification, validation, analysis and evaluation.

This part starts with a paper by M. H. Diallo, L. Naslavsky, T. A. Alspaugh, H. Ziv and D. J. Richardson that is entitled "Toward Architecture Evaluation Through Ontology-Based Requirements-Level Scenario." The paper describes an approach for evaluating whether a candidate architecture dependably satisfies stakeholder requirements expressed in requirements-level scenarios. The approach maps scenarios expressing both functional requirements and quality attributes of the system, to architectural elements through an ontology of requirements-level event classes and domain entities. This in turn provides a clear connection between stakeholder requirements and architectural solutions to address them.

In their paper entitled "Combining Formal Verification and Testing for Correct Legacy Component Integration in Mechatronic UML," H. Giese, S. Henkler and M. Hirsch present a combined use of testing and formal verification for verifying complex real-time component-based systems that include legacy components. The approach is motivated by the need of sufficiently validating the integration of real-time, embedded, and legacy components, and by the claim that a testing phase alone, when applied to such a domain, cannot provide enough guarantees. Thus, the approach proposed here is composed of many steps: a behavioral model of embedded legacy components is derived from the existing interface description; then, such an initial model is submitted to formal verification; if the verification step identifies a failure, the produced counterexample is used to test the legacy component; the execution information is used for refining the behavioral model of the legacy components; the new synthesized behavior is then the starting point for the next iteration. In summary, while formal verification is used for verifying components interactions, local testing of the legacy components is used to refine the behavioral model.

S. Wang, G. S. Avrunin and L. A. Clarke, in their paper "Plug-and-Play Architectural Design and Verification," focus on software plug-and-play architectural design. They propose an approach that allows designers to experiment with alternative design choices of component interactions in a plug-and-play manner. The paper describes how to design and present plug-in-play components using a set of notations to show classified component interactions; furthermore it provides details of how reusable formal models can be created for the connector building blocks. This approach is particularly useful to specify and present different component connection relationships.

In the fourth paper, entitled "Data Flow-based Validation of Web Services Compositions: Perspectives and Examples," C. Bartolini, A. Bertolino, E. Marchetti and I. Parissis describe the use of data flow modeling for testing composite Web services (WSs). The central problem on testing WSs is that the dynamic binding of services makes it impractical to test in advance all the concrete service combinations that can be involved in a workflow. By considering in an explicit way a model of how data are expected to be exchanged between the combined services, it is possible to check whether desired properties are satisfied or also to test whether the implemented Web services composition (WSCs) complies with that model. The authors discuss ways in which, depending on the information available, the flow of data in WSCs can be usefully referred to for verification and validation purposes.

The fifth paper, by T. Kettu, E. Kruse, M. Larsson and G. Mustapic, is entitled "Using Architecture Analysis to Evolve Complex Industrial Systems." This paper from industry provides practical advice on how to reconstruct the architecture of existing software

systems by combining the use of tools and the existing knowledge within the organization. The authors claim that to obtain an up-to-date view of the system and prevent expensive mistakes during system evolution, it is fundamental to obtain an up-to-date view of the architecture of the system. The paper is based on experiences from two cases related to industrial automation.

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 2007 WADS and CompArch 2007 ROSATEA participants for their active participation in the discussions, and Alfred Hofmann from Springer for believing in the idea of a book series on this important topic and for helping us to get it published. Last but not least, we appreciate very much the efforts of our reviewers who helped us in ensuring the high quality of the contributions. They are Sascha Konrad, Tao Xie, Graham Kirby, Rick Kazman, Simona Bernardi, Istvan Majzik, Bedir Tekinerdogan, Miguel Correia, Gergely Pinter, Silvano Chiaradonna, Stephan Storck, Sandro Bologna, Santosh Shrivastava, Paolo Lollini, Alan Hartman, Giovanna Dondossola, Nuno Neves, Sasikumar Punnekkat, Mauro Gaspari, Holger Giese, Eda Marchetti, Kristina Lundqvist, Paris Avgeriou, Luciano Baresi, Andrea Polini, Joseph Slember, Andreas Ulrich, Andrea Bondavalli, Suzanne Embury, Jerry Gao, Roberto Baldoni, Daniel Paulish, Stephan Storck, and several anonymous reviewers.

June 2008

Rogério de Lemos
Felicita Di Giandomenico
Cristina Gacek
Henry Muccini
Marlon Vieira

Foreword

Innovative, high-impact research results in the sciences and engineering may seem to an outsider to have sprung forth from a vacuum, but in truth, they are the result of novel combinations of known ideas. The right conditions to produce such results are often created through the intentional mixing of different communities, each with its own point of view.

For six years now, the Workshop on Architecting Dependable Systems (WADS) has brought together two distinct and very different communities, the software architecture and dependability communities, in order to provide a seedbed for the growth of new ideas concerning the design, construction, and validation of large-scale software-based systems that must be dependable. The importance of bringing together these communities has steadily grown during this period, as our society's dependence on information-technology-based systems continues to grow and as the amount of software in such systems increases.

This volume in the series focuses on methods for designing and validating critical infrastructures, both from an application-driven, top-down perspective and from a bottom-up, technology-driven perspective. In both cases, the authors include people from both the dependability and security community and the software architecture community. The discussion at the associated workshop at the IEEE/IFIP Dependable Systems and Networks meeting was lively, and the revised and expanded papers presented in this volume capture the results of those discussions, and some of the excitement of the exchanges that day.

I applaud Rogério de Lemos, Felicita Di Giandomenico, Cristina Gacek, Henry Muccini, and Marlon Vieira for their work in putting together this volume, and in their long-standing organization of the WADS series. In bringing together people from both the software architecture and dependability communities in a sustained way each year, they are engaging in a community-building effort that could have a significant payoff: the creation of the ability to architect software-based systems that are dependable by design, and remain dependable when configured in different ways throughout their range of use and lifecycle.

Such community-building is hard work, but its value is immense. I look forward to seeing the ongoing efforts that are reflected in this volume bear fruit for many years to come.

June 2008 William H. Sanders

Table of Contents

Part 1. Critical Infrastructures

Part 2. Rigorous Design/Fault Tolerance

Part 3. Verification and Validation