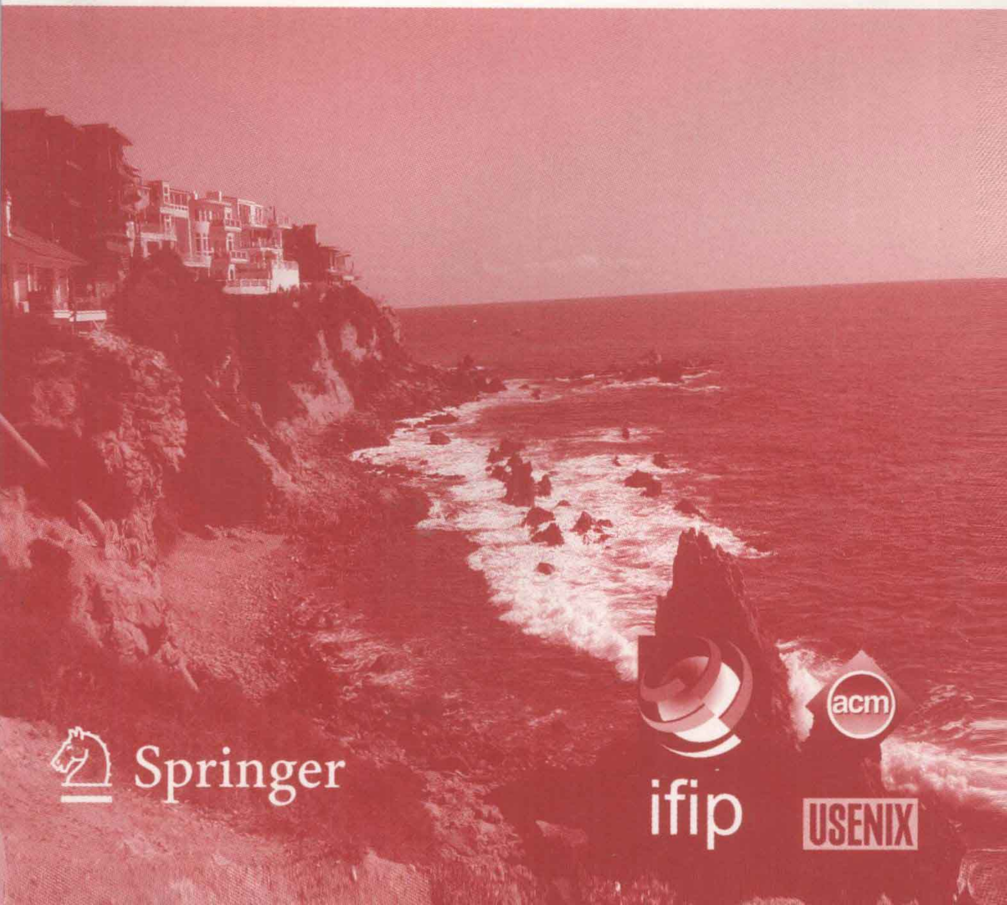
Renato Cerqueira
Roy H. Campbell (Eds.)

LNCS 4834

Middleware 2007

ACM/IFIP/USENIX 8th International Middleware Conference
Newport Beach, CA, USA, November 2007
Proceedings



Renato Cerqueira Roy H. Campbell (Eds.)

Middleware 2007

ACM/IFIP/USENIX
8th International Middleware Conference
Newport Beach, CA, USA, November 26-30, 2007
Proceedings

Springer

Volume Editors

Renato Cerqueira
PUC-Rio, Department of Informatics
Rua Marques de S. Vicente 225, 22453-900 Rio de Janeiro, Brazil
E-mail: rcerq@inf.puc-rio.br

Roy H. Campbell
University of Illinois at Urbana-Champaign
Department of Computer Science, Siebel Center for Computer Science
201 N. Goodwin Avenue MC-258, Urbana, IL 61801-2302, USA
E-mail: rhc@uiuc.edu

Library of Congress Control Number: 2007938898

CR Subject Classification (1998): C.2.4, D.4, C.2, D.1.3, D.3.2, D.2, H.2, H.4

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-76777-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-76777-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12191420 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4834

Preface

Nowadays, middleware technologies are the main infrastructure to support the development and execution of distributed systems, providing design abstractions, programming models and tools, frameworks, protocols, deployment mechanisms, and runtime services. Due to its broad scope, middleware research encompasses different research areas, such as distributed systems, operating systems, networking, multimedia systems, databases, programming languages, and software engineering.

This volume contains the proceedings of the Eighth Middleware Conference, held in Newport Beach, California, USA, November 26–30, 2007. Middleware is a series of conferences that started in 1998 with the aim of being the premier conference on middleware research and technology, where researchers from academia and industry can present and discuss the latest middleware results. The focus of the conference is the design, implementation, deployment, and evaluation of distributed systems platforms and architectures for future computing environments.

This year, we had 108 submissions from 25 different countries, among which the top 22 papers were selected for inclusion in the technical program of the conference. All papers were evaluated by at least three reviewers with respect to their originality, technical merit, presentation quality, and relevance to the conference themes. The selected papers present the latest results and breakthroughs on middleware research in areas including peer-to-peer computing, event-based and publish/subscribe architectures, mobile and ubiquitous systems, grid and cluster computing, sensor networks, component- and Web-based middleware, virtual machines, adaptive and autonomic systems, communication protocols and architectures, scalability, fault-tolerance, quality-of-service, resource management, multimedia streaming, and novel development paradigms and tools.

Middleware 2007 also featured an Experience Papers session, which consisted of papers with focus on applications and experience from the use of middleware. From the research paper submissions, four papers were invited to be presented in this session. Another eight papers were recommended for inclusion in the conference's Work-in-Progress Papers program.

Apart from the papers, the program included seven workshops, a doctoral symposium, invited talks, poster and demo presentations, and panels. We hope that the attendees enjoyed this year's Middleware Conference, gained new knowledge and insights from our program, participated in the presentations and discussions, and met others working on projects similar to theirs.

We would like to express our deepest appreciation to the authors of the submitted papers, to all Program Committee members for their diligence in the paper review and selection process, and to all external reviewers for their help in evaluating submissions. We would also like to thank ACM, IFIP, and USENIX for their technical sponsorship, and the corporate sponsors for their financial support. Finally, special thanks go to Nalini Venkatasubramanian and all the other Organizing Committee members for their hard work and effort to make Middleware 2007 a successful conference.

September 2007

Renato Cerqueira
Roy H. Campbell

Organization

Middleware 2007 was organized under the auspices of IFIP TC6WG6.1 (International Federation for Information Processing, Technical Committee 6 [Communication Systems], Working Group 6.1 [Architecture and Protocols for Computer Networks]).

Executive Committee

Conference Chair	Nalini Venkatasubramanian (University of California, Irvine, USA)
Program Chairs	Roy H. Campbell (University of Illinois, Urbana Champaign, USA)
	Renato Cerqueira (PUC-Rio, Brazil)
Workshop Chairs	Radha Nandkumar (NCSA, USA)
	Baochun Li (University of Toronto, Canada)
Doctoral Symposium Chair	Qi Han (Colorado School of Mines, USA)
Industry Track Chair	Guruduth Banavar (IBM India Research Lab, India)
Work in Progress Chair	Edward Curry (Digital Enterprise Research Institute, Ireland)
Poster and Demo Chair	Valerie Issarny (INRIA, France)
Publicity Chair	Shivajit Mohapatra (Motorola Labs, USA)
Publicity in Europe	Paolo Bellavista (Università degli Studi di Bologna, Italy)
Local Arrangements Chair	Ray Klefstad (University of California, Irvine, USA)
Local Arrangements Committee	Sharad Mehrotra (University of California, Irvine, USA)
	Vana Kalogeraki (University of California, Riverside, USA)
	Deborah Nielsen (OCTANE@UCI, USA)
	Shellie Nazarenus (Calit2, USA)

Steering Committee

Gordon Blair (Chair)	Lancaster University, UK
Gustavo Alonso	ETH Zurich, Switzerland
Jan de Meer	SmartSpaceLab, Germany
Peter Honeyman	CITI, University of Michigan, USA
Hans-Arno Jacobsen	University of Toronto, Canada
Shanika Karunasekera	University of Melbourne, Australia

Elie Najm	ENST, France
Jean-Bernard Stefani	INRIA, France
Joe Sventek	University of Glasgow, UK
Maarten van Steen	Vrije Universiteit Amsterdam, Netherlands
Steve Vinoski	Verivue, USA

Program Committee

Gustavo Alonso	ETH Zurich, Switzerland
Cristiana Amza	University of Toronto, Canada
Jean Bacon	University of Cambridge, UK
Guruduth Banavar	IBM India Research Lab, India
Alberto Bartoli	University of Trieste, Italy
John Barton	IBM Almadem, USA
Christian Becker	University of Mannheim, Germany
Yolande Berbers	Katholieke Universiteit Leuven, Belgium
Gordon Blair	Lancaster University, UK
Jean-Pierre Briot	CNRS and University Paris 6, France
Andrew Campbell	Dartmouth College, USA
Walfredo Cirne	Google, USA
Clem Cole	USENIX, USA
Paolo Costa	Vrije Universiteit Amsterdam, Netherlands
Geoff Coulson	Lancaster University, UK
Jan de Meer	SmartSpaceLab, Germany
Fred Douglis	IBM T.J. Watson Research Center, USA
Naranker Dulay	Imperial College, UK
Markus Endler	PUC-Rio, Brazil
Pascal Felber	University of Neufchatel, Switzerland
Paulo Ferreira	INESC ID / Tech. Univ. of Lisbon, Portugal
Joni Fraga	Federal University of Santa Catarina, Brazil
Indranil Gupta	University of Illinois at Urbana-Champaign, USA
Valerie Issarny	INRIA, France
Hans-Arno Jacobsen	University of Toronto, Canada
Shanika Karunasekera	University of Melbourne, Australia
Bettina Kemme	McGill University, Canada
Fabio Kon	IME/USP, Brazil
Doug Lea	Oswego State University, USA
Rodger Lea	University of British Columbia, Canada
Satoshi Matsuoka	Tokyo Institute of Technology, Japan
Elie Najm	ENST, France
Priya Narasimhan	CMU, USA
Tamer Ozsu	University of Waterloo, Canada
Christian Perez	INRIA/IRISA, France
Gian Pietro Picco	University of Trento, Italy
Alexander Reinefeld	ZIB, Germany

Luis Rodrigues	University of Lisbon, Portugal
Antony Rowstron	Microsoft Research Cambridge, UK
Rick Schantz	BBN Technologies, USA
Douglas C. Schmidt	Vanderbilt University, USA
Jean-Bernard Stefani	INRIA, France
Joe Sventek	University of Glasgow, UK
Stefan Tai	IBM T.J. Watson Research Center, USA
Peter Triantafillou	University of Patras, Greece
Gregor von Laszewski	Argonne National Laboratory, USA

Referees

Ioannis Aekaterinidis
Hossam Afifi
Eduardo Alchieri
Nazareno Andrade
Zahid Anwar
Michael Atighetchi
Pedro Brandão
Raphael Camargo
Fernando Castor Filho
Matteo Ceriotti
Lipo Chan
Ellick Chen
Alex Cheung
Lauro Costa
Marco Danelutto
Khuzaima Daudjee
Mike Dean
Alvaro Degas
Mihai Dobrescu
David Evans
Daniel Fireman
Davide Frey
Matthew Gillen
Alfredo Goldman
Minor Gordon
Paul Grace
Irfan Hamid
Aaron Harwood
Peter Honeyman
Danny Hughes
Roberto Ierusalimschy
Mathieu Jan
Björn Kolbeck
Samuel Kounev
Avraham Leff
João Leitão
Guoli Li
Jian Li
Antónia Lopes
Paulo Mafra
Renato Maia
Olivier Marin
Thomas Mikalsen
Hugo Miranda
José Mocito
Monika Moser
Vinod Muthusamy
Nanjangud Narendra
Nikos Ntarmos
Rafael Obelheiro
Marcelo Iury Oliveira
José A. de Oliveira Neto
Partha Pal
Sujay Parekh
Frédéric Peschanski
Theoni Pitoura
Stefan Plantikow
Michael Rabinovich
Anand Ranganathan
Thomas Röblitz
Kurt Rohloff
Claudia Roncancio
Paul Rubel
Gregor Schiele
Matthias Schunter
Thorsten Schütt
Sabina Serbu
Praveen Sharma
Reza Sherafat
Marc Shiely
Sriram Srinivasan
Biplav Srivastava
Jan Stender
Niko Thio
Muhammad Umer
Luís Veiga
Jan Vitek
Qiang Wang
Michelle Wangham
Matt Welsh
Alex Wun
Eiko Yoneki
Bechir Zalila
Charles Zhang
Ning Zhang
Chenfeng Zhou

Sponsoring Institutions

IFIP (International Federation for Information Processing)
http://www.ifip.org

Association for Computing Machinery
http://www.acm.org

Advanced Computing Systems Association
http://www.usenix.org

Corporate Sponsors

IBM Research
http://www.research.ibm.com

BBN Technologies
http://www.bbn.com

OCTANe
http://www.octaneoc.org

Google
http://www.google.com

Lecture Notes in Computer Science

Sublibrary 2: Programming and Software Engineering

For information about Vols. 1–4192
please contact your bookseller or Springer

Vol. 4834: R. Cerqueira, R.H. Campbell (Eds.), Middleware 2007. XIII, 451 pages. 2007.

Vol. 4824: A. Paschke, Y. Biletskiy (Eds.), Advances in Rule Interchange and Applications. XIII, 243 pages. 2007.

Vol. 4807: Z. Shao (Ed.), Programming Languages and Systems. XI, 431 pages. 2007.

Vol. 4789: M. Butler, M.G. Hinchey, M.M. Larrondo-Petrie (Eds.), Formal Methods and Software Engineering. VIII, 387 pages. 2007.

Vol. 4767: F. Arbab, M. Sirjani (Eds.), International Symposium on Fundamentals of Software Engineering. XIII, 450 pages. 2007.

Vol. 4764: P. Abrahamsson, N. Baddoo, T. Margaria, R. Messnarz (Eds.), Software Process Improvement. XI, 225 pages. 2007.

Vol. 4762: K.S. Namjoshi, T. Yoneda, T. Higashino, Y. Okamura (Eds.), Automated Technology for Verification and Analysis. XIV, 566 pages. 2007.

Vol. 4758: F. Oquendo (Ed.), Software Architecture. XVI, 340 pages. 2007.

Vol. 4757: F. Cappello, T. Herault, J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. XVI, 396 pages. 2007.

Vol. 4753: E. Duval, R. Klamma, M. Wolpers (Eds.), Creating New Learning Experiences on a Global Scale. XII, 518 pages. 2007.

Vol. 4749: B.J. Krämer, K.-J. Lin, P. Narasimhan (Eds.), Service-Oriented Computing – ICSOC 2007. XIX, 629 pages. 2007.

Vol. 4748: K. Wolter (Ed.), Formal Methods and Stochastic Models for Performance Evaluation. X, 301 pages. 2007.

Vol. 4741: C. Bessière (Ed.), Principles and Practice of Constraint Programming – CP 2007. XV, 890 pages. 2007.

Vol. 4735: G. Engels, B. Opdyke, D.C. Schmidt, F. Weil (Eds.), Model Driven Engineering Languages and Systems. XV, 698 pages. 2007.

Vol. 4716: B. Meyer, M. Joseph (Eds.), Software Engineering Approaches for Offshore and Outsourced Development. X, 201 pages. 2007.

Vol. 4680: F. Saglietti, N. Oster (Eds.), Computer Safety, Reliability, and Security. XV, 548 pages. 2007.

Vol. 4670: V. Dahl, I. Niemelä (Eds.), Logic Programming. XII, 470 pages. 2007.

Vol. 4652: D. Georgakopoulos, N. Ritter, B. Benatallah, C. Zirpins, G. Feuerlicht, M. Schoenherr, H.R. Motahari-Nezhad (Eds.), Service-Oriented Computing ICSOC 2006. XVI, 201 pages. 2007.

Vol. 4634: H. Riis Nielson, G. Filé (Eds.), Static Analysis. XI, 469 pages. 2007.

Vol. 4615: R. de Lemos, C. Gacek, A. Romanovsky (Eds.), Architecting Dependable Systems IV. XIV, 435 pages. 2007.

Vol. 4610: B. Xiao, L.T. Yang, J. Ma, C. Muller-Schloer, Y. Hua (Eds.), Autonomic and Trusted Computing. XVIII, 571 pages. 2007.

Vol. 4609: E. Ernst (Ed.), ECOOP 2007 – Object-Oriented Programming. XIII, 625 pages. 2007.

Vol. 4608: H.W. Schmidt, I. Crnković, G.T. Heineman, J.A. Stafford (Eds.), Component-Based Software Engineering. XII, 283 pages. 2007.

Vol. 4591: J. Davies, J. Gibbons (Eds.), Integrated Formal Methods. IX, 660 pages. 2007.

Vol. 4589: J. Münch, P. Abrahamsson (Eds.), Product-Focused Software Process Improvement. XII, 414 pages. 2007.

Vol. 4574: J. Derrick, J. Vain (Eds.), Formal Techniques for Networked and Distributed Systems – FORTE 2007. XI, 375 pages. 2007.

Vol. 4556: C. Stephanidis (Ed.), Universal Access in Human-Computer Interaction, Part III. XXII, 1020 pages. 2007.

Vol. 4555: C. Stephanidis (Ed.), Universal Access in Human-Computer Interaction, Part II. XXII, 1066 pages. 2007.

Vol. 4554: C. Stephanidis (Ed.), Universal Acess in Human Computer Interaction, Part I. XXII, 1054 pages. 2007.

Vol. 4553: J.A. Jacko (Ed.), Human-Computer Interaction, Part IV. XXIV, 1225 pages. 2007.

Vol. 4552: J.A. Jacko (Ed.), Human-Computer Interaction, Part III. XXI, 1038 pages. 2007.

Vol. 4551: J.A. Jacko (Ed.), Human-Computer Interaction, Part II. XXIII, 1253 pages. 2007.

Vol. 4550: J.A. Jacko (Ed.), Human-Computer Interaction, Part I. XXIII, 1240 pages. 2007.

Vol. 4542: P. Sawyer, B. Paech, P. Heymans (Eds.), Requirements Engineering: Foundation for Software Quality. IX, 384 pages. 2007.

Vol. 4536: G. Concas, E. Damiani, M. Scotto, G. Succi (Eds.), Agile Processes in Software Engineering and Extreme Programming. XV, 276 pages. 2007.

Vol. 4530: D.H. Akehurst, R. Vogel, R.F. Paige (Eds.), Model Driven Architecture - Foundations and Applications. X, 219 pages. 2007.

Vol. 4523: Y.-H. Lee, H.-N. Kim, J. Kim, Y.W. Park, L.T. Yang, S.W. Kim (Eds.), Embedded Software and Systems. XIX, 829 pages. 2007.

Vol. 4498: N. Abdennahder, F. Kordon (Eds.), Reliable Software Technologies - Ada-Europe 2007. XII, 247 pages. 2007.

Vol. 4486: M. Bernardo, J. Hillston (Eds.), Formal Methods for Performance Evaluation. VII, 469 pages. 2007.

Vol. 4470: Q. Wang, D. Pfahl, D.M. Raffo (Eds.), Software Process Dynamics and Agility. XI, 346 pages. 2007.

Vol. 4468: M.M. Bonsangue, E.B. Johnsen (Eds.), Formal Methods for Open Object-Based Distributed Systems. X, 317 pages. 2007.

Vol. 4467: A.L. Murphy, J. Vitek (Eds.), Coordination Models and Languages. X, 325 pages. 2007.

Vol. 4454: Y. Gurevich, B. Meyer (Eds.), Tests and Proofs. IX, 217 pages. 2007.

Vol. 4444: T. Reps, M. Sagiv, J. Bauer (Eds.), Program Analysis and Compilation, Theory and Practice. X, 361 pages. 2007.

Vol. 4440: B. Liblit, Cooperative Bug Isolation. XV, 101 pages. 2007.

Vol. 4408: R. Choren, A. Garcia, H. Giese, H.-f. Leung, C. Lucena, A. Romanovsky (Eds.), Software Engineering for Multi-Agent Systems V. XII, 233 pages. 2007.

Vol. 4406: W. De Meuter (Ed.), Advances in Smalltalk. VII, 157 pages. 2007.

Vol. 4405: L. Padgham, F. Zambonelli (Eds.), Agent-Oriented Software Engineering VII. XII, 225 pages. 2007.

Vol. 4401: N. Guelfi, D. Buchs (Eds.), Rapid Integration of Software Engineering Techniques. IX, 177 pages. 2007.

Vol. 4385: K. Coninx, K. Luyten, K.A. Schneider (Eds.), Task Models and Diagrams for Users Interface Design. XI, 355 pages. 2007.

Vol. 4383: E. Bin, A. Ziv, S. Ur (Eds.), Hardware and Software, Verification and Testing. XII, 235 pages. 2007.

Vol. 4379: M. Südholt, C. Consel (Eds.), Object-Oriented Technology. VIII, 157 pages. 2007.

Vol. 4364: T. Kühne (Ed.), Models in Software Engineering. XI, 332 pages. 2007.

Vol. 4355: J. Julliand, O. Kouchnarenko (Eds.), B 2007: Formal Specification and Development in B. XIII, 293 pages. 2006.

Vol. 4354: M. Hanus (Ed.), Practical Aspects of Declarative Languages. X, 335 pages. 2006.

Vol. 4350: M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, C. Talcott, All About Maude - A High-Performance Logical Framework. XXII, 797 pages. 2007.

Vol. 4348: S. Tucker Taft, R.A. Duff, R.L. Brukardt, E. Plödereder, P. Leroy, Ada 2005 Reference Manual. XXII, 765 pages. 2006.

Vol. 4346: L. Brim, B.R. Haverkort, M. Leucker, J. van de Pol (Eds.), Formal Methods: Applications and Technology. X, 363 pages. 2007.

Vol. 4344: V. Gruhn, F. Oquendo (Eds.), Software Architecture. X, 245 pages. 2006.

Vol. 4340: R. Prodan, T. Fahringer, Grid Computing. XXIII, 317 pages. 2007.

Vol. 4336: V.R. Basili, H.D. Rombach, K. Schneider, B. Kitchenham, D. Pfahl, R.W. Selby (Eds.), Empirical Software Engineering Issues. XVII, 193 pages. 2007.

Vol. 4326: S. Göbel, R. Malkewitz, I. Iurgel (Eds.), Technologies for Interactive Digital Storytelling and Entertainment. X, 384 pages. 2006.

Vol. 4323: G. Doherty, A. Blandford (Eds.), Interactive Systems. XI, 269 pages. 2007.

Vol. 4322: F. Kordon, J. Sztipanovits (Eds.), Reliable Systems on Unreliable Networked Platforms. XIV, 317 pages. 2007.

Vol. 4309: P. Inverardi, M. Jazayeri (Eds.), Software Engineering Education in the Modern Age. VIII, 207 pages. 2006.

Vol. 4294: A. Dan, W. Lamersdorf (Eds.), Service-Oriented Computing – ICSOC 2006. XIX, 653 pages. 2006.

Vol. 4290: M. van Steen, M. Henning (Eds.), Middleware 2006. XIII, 425 pages. 2006.

Vol. 4279: N. Kobayashi (Ed.), Programming Languages and Systems. XI, 423 pages. 2006.

Vol. 4262: K. Havelund, M. Núñez, G. Roşu, B. Wolff (Eds.), Formal Approaches to Software Testing and Runtime Verification. VIII, 255 pages. 2006.

Vol. 4260: Z. Liu, J. He (Eds.), Formal Methods and Software Engineering. XII, 778 pages. 2006.

Vol. 4257: I. Richardson, P. Runeson, R. Messnarz (Eds.), Software Process Improvement. XI, 219 pages. 2006.

Vol. 4242: A. Rashid, M. Aksit (Eds.), Transactions on Aspect-Oriented Software Development II. IX, 289 pages. 2006.

Vol. 4229: E. Najm, J.-F. Pradat-Peyre, V.V. Donzeau-Gouge (Eds.), Formal Techniques for Networked and Distributed Systems - FORTE 2006. X, 486 pages. 2006.

Vol. 4227: W. Nejdl, K. Tochtermann (Eds.), Innovative Approaches for Learning and Knowledge Sharing. XVII, 721 pages. 2006.

Vol. 4218: S. Graf, W. Zhang (Eds.), Automated Technology for Verification and Analysis. XIV, 540 pages. 2006.

Vol. 4214: C. Hofmeister, I. Crnković, R. Reussner (Eds.), Quality of Software Architectures. X, 215 pages. 2006.

Vol. 4204: F. Benhamou (Ed.), Principles and Practice of Constraint Programming - CP 2006. XVIII, 774 pages. 2006.

Vol. 4199: O. Nierstrasz, J. Whittle, D. Harel, G. Reggio (Eds.), Model Driven Engineering Languages and Systems. XVI, 798 pages. 2006.

Table of Contents

Component-Based Middleware

Mobile and Ubiquitous Computing

Grid and Cluster Computing

Enhancing Communication

Resource Management

Reliability and Fault Tolerance

Asynchronous Communication

R-OSGi: Distributed Applications Through Software Modularization

Jan S. Rellermeyer, Gustavo Alonso, and Timothy Roscoe

Department of Computer Science, ETH Zurich
8092 Zurich, Switzerland
{rellermeyer, alonso, troscoe}@inf.ethz.ch

Abstract. In this paper we take advantage of the concepts developed for centralized module management, such as dynamic loading and unloading of modules, and show how they can be used to support the development and deployment of distributed applications. We do so through R-OSGi, a distributed middleware platform that extends the centralized, industry-standard OSGi specification to support distributed module management. To the developer, R-OSGi looks like a conventional module management tool. However, at deployment time, R-OSGi can be used to turn the application into a distributed application by simply indicating where the different modules should be deployed. At run time, R-OSGi represents distributed failures as module insertion and withdrawal operations so that the logic to deal with failures is the same as that employed to deal with dependencies among software modules. In doing so, R-OSGi greatly simplifies the development of distributed applications with no performance cost. In the paper we describe R-OSGi and several use cases. We also show with extensive experiments that R-OSGi has a performance comparable or better than that of RMI or UPnP, both commonly used distribution mechanisms with far less functionality than R-OSGi.

1 Introduction

Modular design is a cornerstone of software engineering, and much effort has been invested in concepts and tools to manage modules and the dependencies among them. Nowadays, modularization pervades programming languages, development environments, and even system architectures. In particular, recent years have seen the emergence of "module management systems" which handle loading and unloading of modular program units at runtime, and dynamically creating and destroying bindings between services in different modules.

In this paper we explore using centralized module management as the basis for the design and deployment of distributed applications. Our work is based on the OSGi specification [1], a widely used module management API designed to work on a single system that we extend extend to work in a distributed setting.

R. Cerqueira and R.H. Campbell (Eds.): Middleware 2007, LNCS 4834, pp. 1–20, 2007.

The key insight is that the module boundaries instituted by centralized module management systems are generally well-suited to being repurposed as distribution boundaries. In the past, networked applications have typically distributed their functionality by interposing communication proxies at procedure calls or object method invocation, with mixed results. In particular, the issue of transparency has dogged distributed computing platforms based on these models: as Waldo et. al. [2] point out, a remote procedure invocation has fundamentally different semantics to a local call, and consequently fundamentally different exception handling code must be written by the programmer.

In contrast, module management systems like OSGi are designed to handle unloading of modules at any time, and include event notification functionality to enable programmers (indeed, to require them) to cleanly handle services disappearing without notice. We take advantage of this by representing communication-related failures as local module unloading events.

By doing so, we effectively turn software modules into the potential units of distribution. The result is *Remoting-OSGi* (R-OSGi), a distributed middleware platform that can transparently distribute parts of an application by simply distributing its software modules. R-OSGi is a middleware layer on top of OSGi. This matches the lightweight design of OSGi and allows us to use R-OSGi on any OSGi enabled application.

R-OSGi makes the following contributions:

1. *Seamless embedding in OSGi*: From the OSGi framework's point of view, local and remote services are indistinguishable. Existing OSGi applications can be distributed using R-OSGi without modification.
2. *Reliability*: The distribution of services does not add new failure patterns to an OSGi application. Developers deal with network-related errors in the same way they deal with errors caused by module interaction.
3. *Generality*: The middleware is not tailored to a subset of potential services. Every valid OSGi service is potentially accessible by remote peers.
4. *Portability*: The middleware runs Java VM implementations for typical resource-constrained mobile devices, such as PDAs or smartphones. The resource requirements of R-OSGi are by design modest.
5. *Adaptivity*: R-OSGi does not impose role assignments (e.g., client or server). The relation between modules is generally symmetric and so is the distributed application generated by R-OSGi.
6. *Efficiency*: R-OSGi is fast, its performance is comparable to the (highly optimized) Java 5 RMI implementation, and is two orders of magnitude faster than UPnP.

In the next section we discuss in more detail the relevance of module management systems for distributed applications, using OSGi as a case study. In Section 4 we discuss the architecture and design of R-OSGi, and describe the implementation in detail in Section 5. Section 6 presents performance results for R-OSGi, and Section 7 details several use cases including ubiquitous computing