Luciano Baresi
Reiko Heckel (Eds.)

LNCS 3922

Fundamental Approache to Software Engineering

9th International Conference, FASE 2006
Held as Part of the Joint European Conferences
on Theory and Practice of Software, ETAPS 2006
Vienna, Austria, March 2006, Proceedings

European Joint Conferences on
Theory
And
Practice of
Software
2006

Springer

Luciano Baresi   Reiko Heckel (Eds.)

Fundamental Approaches to Software Engineering

9th International Conference, FASE 2006
Held as Part of the Joint European Conferences
on Theory and Practice of Software, ETAPS 2006
Vienna, Austria, March 27-28, 2006
Proceedings

Springer

Volume Editors

Luciano Baresi
Politecnico di Milano, Dipartimento di Elettronica e Informazione
piazza Leonardo da Vinci, 32 - 20133 Milano, Italy
E-mail: baresi@elet.polimi.it

Reiko Heckel
University of Leicester, Department of Computer Science
University Road, LE1 7RH Leicester, UK
E-mail: reiko@mcs.le.ac.uk

Library of Congress Control Number: 2006922237

CR Subject Classification (1998): D.2, F.3, D.3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN	0302-9743
ISBN-10	3-540-33093-3 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-33093-6 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: 11693017 06/3142 5 4 3 2 1 0

Foreword

ETAPS 2006 was the ninth instance of the European Joint Conferences on Theory and Practice of Software. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprised five conferences (CC, ESOP, FASE, FOSSACS, TACAS), 18 satellite workshops (ACCAT, AVIS, CMCS, COCV, DCC, EAAI, FESCA, FRCSS, GT-VMT, LDTA, MBT, QAPL, SC, SLAP, SPIN, TERMGRAPH, WITS and WRLA), two tutorials, and seven invited lectures (not including those that were specific to the satellite events). We received over 550 submissions to the five conferences this year, giving an overall acceptance rate of 23%, with acceptance rates below 30% for each conference. Congratulations to all the authors who made it to the final programme! I hope that most of the other authors still found a way of participating in this exciting event and I hope you will continue submitting.

The events that comprise ETAPS address various aspects of the system development process, including specification, design, implementation, analysis and improvement. The languages, methodologies and tools which support these activities are all well within its scope. Different blends of theory and practice are represented, with an inclination towards theory with a practical motivation on the one hand and soundly based practice on the other. Many of the issues involved in software design apply to systems in general, including hardware systems, and the emphasis on software is not intended to be exclusive.

ETAPS is a loose confederation in which each event retains its own identity, with a separate Program Committee and proceedings. Its format is open-ended, allowing it to grow and evolve as time goes by. Contributed talks and system demonstrations are in synchronized parallel sessions, with invited lectures in plenary sessions. Two of the invited lectures are reserved for "unifying" talks on topics of interest to the whole range of ETAPS attendees. The aim of cramming all this activity into a single one-week meeting is to create a strong magnet for academic and industrial researchers working on topics within its scope, giving them the opportunity to learn about research in related areas, and thereby to foster new and existing links between work in areas that were formerly addressed in separate meetings.

ETAPS 2006 was organized by the Vienna University of Technology, in cooperation with:

- European Association for Theoretical Computer Science (EATCS);
- European Association for Programming Languages and Systems (EAPLS);
- European Association of Software Science and Technology (EASST);
- Institute for Computer Languages, Vienna;
- Austrian Computing Society;
- The *Bürgermeister der Bundeshauptstadt Wien*;
- Vienna Convention Bureau;
- Intel.

The organizing team comprised:

Chair:	Jens Knoop
Local Arrangements:	Anton Ertl
Publicity:	Joost-Pieter Katoen
Satellite Events:	Andreas Krall
Industrial Liaison:	Eva Kühn
Liaison with City of Vienna:	Ulrich Neumerkel
Tutorials Chair, Website:	Franz Puntigam
Website:	Fabian Schmied
Local Organization, Workshops Proceedings:	Markus Schordan

Overall planning for ETAPS conferences is the responsibility of its Steering Committee, whose current membership is:

Perdita Stevens (Edinburgh, Chair), Luca Aceto (Aalborg and Reykjavík), Rastislav Bodík (Berkeley), Maura Cerioli (Genova), Matt Dwyer (Nebraska), Hartmut Ehrig (Berlin), José Fiadeiro (Leicester), Marie-Claude Gaudel (Paris), Roberto Gorrieri (Bologna), Reiko Heckel (Leicester), Michael Huth (London), Joost-Pieter Katoen (Aachen), Paul Klint (Amsterdam), Jens Knoop (Vienna), Shriram Krishnamurthi (Brown), Kim Larsen (Aalborg), Tiziana Margaria (Göttingen), Ugo Montanari (Pisa), Rocco de Nicola (Florence), Hanne Riis Nielson (Copenhagen), Jens Palsberg (UCLA), Mooly Sagiv (Tel-Aviv), João Saraiva (Minho), Don Sannella (Edinburgh), Vladimiro Sassone (Southampton), Helmut Seidl (Munich), Peter Sestoft (Copenhagen), Andreas Zeller (Saarbrücken).

I would like to express my sincere gratitude to all of these people and organizations, the Program Committee chairs and PC members of the ETAPS conferences, the organizers of the satellite events, the speakers themselves, the many reviewers, and Springer for agreeing to publish the ETAPS proceedings. Finally, I would like to thank the Organizing Chair of ETAPS 2006, Jens Knoop, for arranging for us to have ETAPS in the beautiful city of Vienna.

Edinburgh, January 2006

Perdita Stevens
ETAPS Steering Committee Chair

Preface

Software engineering aims to create a feedback cycle between academia and industry, proposing new solutions and identifying those that "work" in practical contexts. The conference on Fundamental Approaches to Software Engineering (FASE) —as one of the European Joint Conferences on Theory and Practice of Software (ETAPS)— is committed to this aim.

With the society increasingly relying on software, the ability to produce low-cost and high-quality software systems is crucial to technological and social progress. FASE provides software engineers with a forum for discussing theories, languages, methods, and tools derived from the interaction of academic research and real-world experience.

Contributions were sought targeting problems of practical relevance through fundamental contributions, based on solid mathematical or conceptual foundations, which could lead to improved engineering practices.

The response of the scientific community was overwhelming, with record submission numbers of 166 research papers and 7 tool papers. From these, 27 research papers and 2 tool papers were selected for publication, with an overall acceptance rate of 17%. The international character of the conference is underlined by the fact that just about one third of the authors are from European countries, while the others come from North America, Asia and Australia.

Accepted papers address topics like distributed and service-oriented computing, measurement and empirical software engineering, methods and tools for software development, validation and verification, model-based development, and software evolution. The scientific programme is complemented by the invited lecture of Francisco Curbera on "A Programming Model for Service Oriented Applications" and of Carlo Ghezzi on "Software Engineering: Emerging Goals and Lasting Problems".

We are deeply indebted to the 24 members of the Program Committee and the 123 additional reviewers for their invaluable time, spent reading and discussing a large number of papers and producing more then 500 reviews.

FASE 2006 was held in Vienna (Austria), hosted and organized by the Institute for Computer Languages at the Vienna University of Technology. Next year FASE will take place in Braga (Portugal). Being part of ETAPS, FASE shares the sponsoring and support acknowledged in the foreword. Heartfelt thanks are also due to Perdita Stevens for excellent and efficient global coordination and to Jens Knoop and his staff for their wonderful job as local organizers.

Finally, special thanks to all contributors and participants who, at the end of the day, are what this is all about.

Milano and Leicester, January 2006

Luciano Baresi
Reiko Heckel

Organization

Program Committee

Jan Øyvind Aagedal (SINTEF, Oslo, Norway)
Luciano Baresi (Politecnico di Milano, Italy) Co-chair
Jean Bézivin (University of Nantes, France)
Victor Braberman (University of Buenos Aires, Argentina)
Maura Cerioli (Università di Genova, Italy)
Matt Dwyer (University of Nebraska, USA)
Anthony Finkelstein (University College London, UK)
Harald Gall (University of Zurich, Switzerland)
Alan Hartman (IBM, Israel)
Reiko Heckel (University of Leicester, UK) Co-chair
Mehdi Jazayeri (University of Lugano, Switzerland and University of Vienna, Austria)
Antónia Lopes (University of Lisbon, Portugal)
Sandro Morasca (Università dell'Insubria, Italy)
András Pataricza (Budapest University of Technology and Economics, Hungary)
Mauro Pezzè (Università di Milano-Bicocca, Italy)
Arend Rensink (University of Twente, The Netherlands)
Leila Ribeiro (Federal University of Rio Grande do Sul, Brazil)
Andy Schürr (University of Darmstadt, Germany)
Gabi Taentzer (Technische Universität Berlin, Germany)
Tetsuo Tamai (University of Tokyo, Japan)
Sebastian Uchitel (Imperial College, UK)
Heike Wehrheim (University of Paderborn, Germany)
Michel Wermelinger (Open University, UK)
Alexander Wolf (University of Lugano, Switzerland and University of Colorado, USA)
Michal Young (University of Oregon, USA)

Referees

N. Aguirre
F. Asteasuain
L. Balmelli
A. Balogh
L. Barroca
A. Benczur
B. Braatz
C. Braghin
I. Brueckner
A. Capiluppi
W. Cazzola
A. Cherchago
J. Cook
S. Costa
E. Cota
I. Craggs
G. Csertán
J. de Lara
H. Dierks
G. Dodero
F. Dotti
S. Dustdar
K. Ehrig
C. Eisner
C. Ermel
J. Faber
E. Farchi
J. Fernandez-Ramil
M. Fischer
A. Fiuri

B. Fluri
L. Foss
M. Frias
D. Garbervetsky
H. Giese
L. Gönczy
R. Grønmo
T. Gyimóthy
J. Hall
J.H. Hausmann
J. Henkel
F. Herrmann
S. Holland
M. Huhn
N. Kicillof
E. Kirda
A. Kirshin
M. Klein
P. Knab
A.R. Kniep
M. Koch
B. Koenig
P. Kosiuczenko
S. Kremer-Davidson
J. Kuester
G. Lagorio
G. Lajios
L. Lamb
L. Lambers
R. Lanotte
M. Lohmann
A.P. Ludtke Ferreira
G. Lukácsy
M. Lund
I. Majzik
K. Mehner
S. Meier
H. Melgratti
B. Metzler
R. Meyer
M. Mikic-Rakic
S. Minocha
T. Modica
M. Moeller
M. Monga
D. Monteverde
Á. Moreira
A. Murphy
J. Oberleitner
J. Oldevik
A. Olivero
S. Olvovsky
J. Padberg
M. Pavlovic
G. Pintér
M.C. Pinto
M. Pinzger
A. Platzer
U. Prange
E. Quintarelli
R. Quites Reis
L. Rabelo
O.W. Rahlff
G. Rangel
L. Rapanotti
H. Rasch
A. Rausch
G. Reggio
G. Reif
R. Reussner
M. Ribaudo
J. Rubin
L. Samuelis
A. Schaefer
F. Schapachnik
Y. Shaham-Gafny
N. Sharygina
A. Solberg
J. Stafford
T. Staijen
S. Sutton
G. Toffetti Carughi
A. Toval
S. Ur
M. Urtasun
J. van der Linden
M. van der Meulen
G. van Dijk
D. Varró
G. Varró
J. Whitehead
S. Yovine
E. Zucca

Lecture Notes in Computer Science 3922

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

Lecture Notes in Computer Science

For information about Vols. 1–3815

please contact your bookseller or Springer

Vol. 3925: A. Valmari (Ed.), Model Checking Software. X, 307 pages. 2006.

Vol. 3924: P. Sestoft (Ed.), Programming Languages and Systems. XII, 343 pages. 2006.

Vol. 3923: A. Mycroft, A. Zeller (Eds.), Compiler Construction. XV, 277 pages. 2006.

Vol. 3922: L. Baresi, R. Heckel (Eds.), Fundamental Approaches to Software Engineering. XIII, 427 pages. 2006.

Vol. 3921: L. Aceto, A. Ingólfsdóttir (Eds.), Foundations of Software Science and Computational Structures. XV, 447 pages. 2006.

Vol. 3920: H. Hermanns, J. Palsberg (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XIV, 506 pages. 2006.

Vol. 3916: J. Li, Q. Yang, A.-H. Tan (Eds.), Data Mining for Biomedical Applications. VIII, 155 pages. 2006. (Sublibrary LNBI).

Vol. 3904: M. Baldoni, U. Endriss, A. Omicini, P. Torroni (Eds.), Declarative Agent Languages and Technologies III. XII, 245 pages. 2006. (Sublibrary LNAI).

Vol. 3903: K. Chen, R. Deng, X. Lai, J. Zhou (Eds.), Information Security Practice and Experience. XIV, 392 pages. 2006.

Vol. 3901: P.M. Hill (Ed.), Logic Based Program Synthesis and Transformation. X, 179 pages. 2006.

Vol. 3899: S. Frintrop, VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search. XIV, 216 pages. 2006. (Sublibrary LNAI).

Vol. 3896: Y. Ioannidis, M.H. Scholl, J.W. Schmidt, F. Matthes, M. Hatzopoulos, K. Boehm, A. Kemper, T. Grust, C. Boehm (Eds.), Advances in Database Technology - EDBT 2006. XIV, 1208 pages. 2006.

Vol. 3895: O. Goldreich, A.L. Rosenberg, A.L. Selman (Eds.), Theoretical Computer Science. XII, 399 pages. 2006.

Vol. 3894: W. Grass, B. Sick, K. Waldschmidt (Eds.), Architecture of Computing Systems - ARCS 2006. XII, 496 pages. 2006.

Vol. 3890: S.G. Thompson, R. Ghanea-Hercock (Eds.), Defence Applications of Multi-Agent Systems. XII, 141 pages. 2006. (Sublibrary LNAI).

Vol. 3889: J. Rosca, D. Erdogmus, J.C. Príncipe, S. Haykin (Eds.), Independent Component Analysis and Blind Signal Separation. XXI, 980 pages. 2006.

Vol. 3888: D. Draheim, G. Weber (Eds.), Trends in Enterprise Application Architecture. IX, 145 pages. 2006.

Vol. 3887: J. Correa, A. Hevia, M. Kiwi (Eds.), LATIN 2006: Theoretical Informatics. XVI, 814 pages. 2006.

Vol. 3886: E.G. Bremer, J. Hakenberg, E.-H.(S.) Han, D. Berrar, W. Dubitzky (Eds.), Knowledge Discovery in Life Science Literature. XIV, 147 pages. 2006. (Sublibrary LNBI).

Vol. 3885: V. Torra, Y. Narukawa, A. Valls, J. Domingo-Ferrer (Eds.), Modeling Decisions for Artificial Intelligence. XII, 374 pages. 2006. (Sublibrary LNAI).

Vol. 3884: B. Durand, W. Thomas (Eds.), STACS 2006. XIV, 714 pages. 2006.

Vol. 3881: S. Gibet, N. Courty, J.-F. Kamp (Eds.), Gesture in Human-Computer Interaction and Simulation. XIII, 344 pages. 2006. (Sublibrary LNAI).

Vol. 3880: A. Rashid, M. Aksit (Eds.), Transactions on Aspect-Oriented Software Development I. IX, 335 pages. 2006.

Vol. 3879: T. Erlebach, G. Persinao (Eds.), Approximation and Online Algorithms. X, 349 pages. 2006.

Vol. 3878: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVII, 589 pages. 2006.

Vol. 3877: M. Detyniecki, J.M. Jose, A. Nürnberger, C. J. '. van Rijsbergen (Eds.), Adaptive Multimedia Retrieval: User, Context, and Feedback. XI, 279 pages. 2006.

Vol. 3876: S. Halevi, T. Rabin (Eds.), Theory of Cryptography. XI, 617 pages. 2006.

Vol. 3875: S. Ur, E. Bin, Y. Wolfsthal (Eds.), Haifa Verification Conference. X, 265 pages. 2006.

Vol. 3874: R. Missaoui, J. Schmidt (Eds.), Formal Concept Analysis. X, 309 pages. 2006. (Sublibrary LNAI).

Vol. 3873: L. Maicher, J. Park (Eds.), Charting the Topic Maps Research and Applications Landscape. VIII, 281 pages. 2006. (Sublibrary LNAI).

Vol. 3872: H. Bunke, A. L. Spitz (Eds.), Document Analysis Systems VII. XIII, 630 pages. 2006.

Vol. 3870: S. Spaccapietra, P. Atzeni, W.W. Chu, T. Catarci, K.P. Sycara (Eds.), Journal on Data Semantics V. XIII, 237 pages. 2006.

Vol. 3869: S. Renals, S. Bengio (Eds.), Machine Learning for Multimodal Interaction. XIII, 490 pages. 2006.

Vol. 3868: K. Römer, H. Karl, F. Mattern (Eds.), Wireless Sensor Networks. XI, 342 pages. 2006.

Vol. 3866: T. Dimitrakos, F. Martinelli, P.Y.A. Ryan, S. Schneider (Eds.), Formal Aspects in Security and Trust. X, 259 pages. 2006.

Vol. 3865: W. Shen, K.-M. Chao, Z. Lin, J.-P.A. Barthès (Eds.), Computer Supported Cooperative Work in Design II. XII, 359 pages. 2006.

Vol. 3863: M. Kohlhase (Ed.), Mathematical Knowledge Management. XI, 405 pages. 2006. (Sublibrary LNAI).

Vol. 3862: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 267 pages. 2006. (Sublibrary LNAI).

Vol. 3861: J. Dix, S.J. Hegner (Eds.), Foundations of Information and Knowledge Systems. X, 331 pages. 2006.

Vol. 3860: D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006. XI, 365 pages. 2006.

Vol. 3858: A. Valdes, D. Zamboni (Eds.), Recent Advances in Intrusion Detection. X, 351 pages. 2006.

Vol. 3857: M.P.C. Fossorier, H. Imai, S. Lin, A. Poli (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. XI, 350 pages. 2006.

Vol. 3855: E. A. Emerson, K.S. Namjoshi (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 443 pages. 2005.

Vol. 3854: I. Stavrakakis, M. Smirnov (Eds.), Autonomic Communication. XIII, 303 pages. 2006.

Vol. 3853: A.J. Ijspeert, T. Masuzawa, S. Kusumoto (Eds.), Biologically Inspired Approaches to Advanced Information Technology. XIV, 388 pages. 2006.

Vol. 3852: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision – ACCV 2006, Part II. XXXI, 977 pages. 2006.

Vol. 3851: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision – ACCV 2006, Part I. XXXI, 973 pages. 2006.

Vol. 3850: R. Freund, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 371 pages. 2006.

Vol. 3849: I. Bloch, A. Petrosino, A.G.B. Tettamanzi (Eds.), Fuzzy Logic and Applications. XIV, 438 pages. 2006. (Sublibrary LNAI).

Vol. 3848: J.-F. Boulicaut, L. De Raedt, H. Mannila (Eds.), Constraint-Based Mining and Inductive Databases. X, 401 pages. 2006. (Sublibrary LNAI).

Vol. 3847: K.P. Jantke, A. Lunzer, N. Spyratos, Y. Tanaka (Eds.), Federation over the Web. X, 215 pages. 2006. (Sublibrary LNAI).

Vol. 3846: H. J. van den Herik, Y. Björnsson, N.S. Netanyahu (Eds.), Computers and Games. XIV, 333 pages. 2006.

Vol. 3845: J. Farré, I. Litovsky, S. Schmitz (Eds.), Implementation and Application of Automata. XIII, 360 pages. 2006.

Vol. 3844: J.-M. Bruel (Ed.), Satellite Events at the MoDELS 2005 Conference. XIII, 360 pages. 2006.

Vol. 3843: P. Healy, N.S. Nikolov (Eds.), Graph Drawing. XVII, 536 pages. 2006.

Vol. 3842: H.T. Shen, J. Li, M. Li, J. Ni, W. Wang (Eds.), Advanced Web and Network Technologies, and Applications. XXVII, 1057 pages. 2006.

Vol. 3841: X. Zhou, J. Li, H.T. Shen, M. Kitsuregawa, Y. Zhang (Eds.), Frontiers of WWW Research and Development - APWeb 2006. XXIV, 1223 pages. 2006.

Vol. 3840: M. Li, B. Boehm, L.J. Osterweil (Eds.), Unifying the Software Process Spectrum. XVI, 522 pages. 2006.

Vol. 3839: J.-C. Filliâtre, C. Paulin-Mohring, B. Werner (Eds.), Types for Proofs and Programs. VIII, 275 pages. 2006.

Vol. 3838: A. Middeldorp, V. van Oostrom, F. van Raamsdonk, R. de Vrijer (Eds.), Processes, Terms and Cycles: Steps on the Road to Infinity. XVIII, 639 pages. 2005.

Vol. 3837: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks. IX, 307 pages. 2005.

Vol. 3836: J.-M. Pierson (Ed.), Data Management in Grids. X, 143 pages. 2006.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005. (Sublibrary LNAI).

Vol. 3834: D.G. Feitelson, E. Frachtenberg, L. Rudolph, U. Schwiegelshohn (Eds.), Job Scheduling Strategies for Parallel Processing. VIII, 283 pages. 2005.

Vol. 3833: K.-J. Li, C. Vangenot (Eds.), Web and Wireless Geographical Information Systems. XI, 309 pages. 2005.

Vol. 3832: D. Zhang, A.K. Jain (Eds.), Advances in Biometrics. XX, 796 pages. 2005.

Vol. 3831: J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, J. Štuller (Eds.), SOFSEM 2006: Theory and Practice of Computer Science. XV, 576 pages. 2006.

Vol. 3830: D. Weyns, H. V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems II. VIII, 291 pages. 2006. (Sublibrary LNAI).

Vol. 3829: P. Pettersson, W. Yi (Eds.), Formal Modeling and Analysis of Timed Systems. IX, 305 pages. 2005.

Vol. 3828: X. Deng, Y. Ye (Eds.), Internet and Network Economics. XVII, 1106 pages. 2005.

Vol. 3827: X. Deng, D.-Z. Du (Eds.), Algorithms and Computation. XX, 1190 pages. 2005.

Vol. 3826: B. Benatallah, F. Casati, P. Traverso (Eds.), Service-Oriented Computing - ICSOC 2005. XVIII, 597 pages. 2005.

Vol. 3824: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embedded and Ubiquitous Computing – EUC 2005. XXIII, 1204 pages. 2005.

Vol. 3823: T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, L.T. Yang (Eds.), Embedded and Ubiquitous Computing – EUC 2005 Workshops. XXXII, 1317 pages. 2005.

Vol. 3822: D. Feng, D. Lin, M. Yung (Eds.), Information Security and Cryptology. XII, 420 pages. 2005.

Vol. 3821: R. Ramanujam, S. Sen (Eds.), FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science. XIV, 566 pages. 2005.

Vol. 3820: L.T. Yang, X.-s. Zhou, W. Zhao, Z. Wu, Y. Zhu, M. Lin (Eds.), Embedded Software and Systems. XXVIII, 779 pages. 2005.

Vol. 3819: P. Van Hentenryck (Ed.), Practical Aspects of Declarative Languages. X, 231 pages. 2005.

Vol. 3818: S. Grumbach, L. Sui, V. Vianu (Eds.), Advances in Computer Science – ASIAN 2005. XIII, 294 pages. 2005.

Vol. 3817: M. Faundez-Zanuy, L. Janer, A. Esposito, A. Satue-Villar, J. Roure, V. Espinosa-Duro (Eds.), Nonlinear Analyses and Algorithms for Speech Processing. XII, 380 pages. 2006. (Sublibrary LNAI).

Vol. 3816: G. Chakraborty (Ed.), Distributed Computing and Internet Technology. XXI, 606 pages. 2005.

¥479.00元

Table of Contents

Invited Contributions

Distributed Systems

Orthogonal Process Activities

Behavioral Models and State Machines

Empirical Studies

Requirements and Design

Model-Based Development

Validation and Verification

Tool Demonstrations

Software Evolution

A Programming Model for Service Oriented Applications

Francisco Curbera

IBM Research, USA
curbera@us.ibm.com

Service oriented computing (SOC) and service oriented architectures introduce a model for distributed software components. Full inter-component interoperability, based on Web services standards, is a core assumption of the SOC model. SOC, as a platform independent approach to software development and management, is not limited to a particular distributed computing stack (Web services), since the benefits of a distributed component model extend to legacy protocols and platforms as well. Web services has successfully stressed the notion that implementation characteristics should be decoupled from interoperability concerns, and has focused on defining an XML based interoperability stack. SOC is directly concerned with the implementation and management of service oriented applications and stresses the ability to incorporate multiple runtimes and programming models into an architecture of distributed software components.

The Service Component Architecture (SCA) is the first realization of SOC as an explicit component model. Just as and Web Services provide the common abstraction of interoperability concerns, SCA provides a common abstraction of implementation concerns. SCA introduces a common notion of service components, service types and service implementations as well as an assembly model for service oriented applications. SCA's goal is to be able to accommodate multiple implementation platforms into a single set of component oriented abstractions. J2EE, BPEL4WS, COBOL, SQL or XML components are only part of the possible implementation artifacts that SCA intends to support. Portability of component assemblies and implementations is an important concern of SCA. SCA is already backed by a Java open source initiative in Apache.

An initiative so ambitious necessarily raises many open issues. Foremost among them is the formalization of an SCA runtime model sufficiently complete to ensure portability of implementations, but at the same time generic enough that it can be supported by multiple platforms and programming models. Once an SCA runtime model is defined, the question arises of whether a "native SCA" platform would be able to provide better support for the execution and deployment of SOC applications. Other significant issues include the possibility of formalizing the component and assembly models beyond their current state, and the support for non-functional requirements and capabilities in the definition and assembly of components.

This talk will review the motivation and major elements of the SCA model, and will discuss the main open issues surrounding the SCA effort.

L. Baresi and R. Heckel (Eds.): FASE 2006, LNCS 3922, p. 1, 2006.

Software Engineering: Emerging Goals and Lasting Problems

Carlo Ghezzi

Dipartimento di Elettronica e Informazione - Politecnico di Milano,
Piazza L. da Vinci 32, I-20133 Milano, Italy
carlo.ghezzi@polimi.it

Software has been evolving from pre-defined, monolithic, centralized architectures to increasingly decentralized, distributed, dynamically composed federations of components. Software processes have been evolving along similar lines, from pre-specified sequential work- flows to decentralized and multi-organization endeavors. The organizations to which software solutions are targeted have also been evolving from highly structured corporates to agile and networked enterprises. All this is affecting the way software is engineered (i.e., conceived, architected, and produced). New difficult challenges arise, while old fundamental problems are still with us. The talk surveys this evolution and tries to identify achievements, challenges, and research directions.

L. Baresi and R. Heckel (Eds.): FASE 2006, LNCS 3922, p. 2, 2006.

GPSL: A Programming Language for Service Implementation

Dominic Cooney, Marlon Dumas, and Paul Roe

Queensland University of Technology, Australia
{d.cooney, m.dumas, p.roe}@qut.edu.au

Abstract. At present, there is a dichotomy of approaches to supporting web service implementation: extending mainstream programming languages with libraries and metadata notations vs. designing new languages. While the former approach has proven suitable for interconnecting services on a simple point-to-point fashion, it turns to be unsuitable for coding concurrent, multi-party, and interrelated interactions requiring extensive XML manipulation. As a result, various web service programming languages have been proposed, most notably (WS-)BPEL. However, these languages still do not meet the needs of highly concurrent and dynamic interactions due to their bias towards statically-bounded concurrency. In this paper we introduce a new web service programming language with a set of features designed to address this gap. We describe the implementations in this language of non-trivial scenarios of service interaction and contrast them to the corresponding BPEL implementations. We also define a formal semantics for the language by translation to the join calculus. A compiler for the language has been implemented based on this semantics.

1 Introduction

There is an increasing acceptance of Service-Oriented Architectures as a paradigm for software application integration. In this paradigm, independently developed and operated applications are exposed as (web) services that are then interconnected using standard protocols and languages [1]. While the technology for developing basic services and interconnecting them on a point-to-point basis has attained some maturity, there remain open challenges when it comes to implementing service interactions that go beyond simple sequences of requests and responses or that involve many participants.

A number of recent and ongoing initiatives aim at tackling these challenges. These initiatives can be classified into conservative extensions to mainstream programming languages and novel service-oriented programming languages. The former provide metadata-based extensions for web service development on top of object-oriented programming languages. For example Microsoft Web Services Extensions, Windows Communication Foundation, Apache Axis and JSR-181, can be placed in this category. While these extensions are suitable for dealing with bilateral interactions and simple forms of concurrency and correlation,

L. Baresi and R. Heckel (Eds.): FASE 2006, LNCS 3922, pp. 3–17, 2006.