LNCS 2949

Rocco De Nicola
Gianluigi Ferrari
Greg Meredith (Eds.)

Coordination Models and Languages

6th International Conference, COORDINATION 2004
Pisa, Italy, February 2004
Proceedings

Springer

Rocco De Nicola Gianluigi Ferrari
Greg Meredith (Eds.)

Coordination Models and Languages

6th International Conference, COORDINATION 2004
Pisa Italy, February 24-27, 2004
Proceedings

Springer

Volume Editors

Rocco De Nicola
Università. di Firenze, Dipartimento di Sistemi e Informatica
Via Lombroso 6/17, 50134 Firenze, Italy
E-mail: denicola@dsi.unifi.it

Gianluigi Ferrari
Università di Pisa, Dipartimento di Informatica
Via F. Buonarroti, 2, 56127 Pisa, Italy
E-mail: giangi@di.unipi.it

Greg Meredith
Microsoft Corporation
Redmont, WA 98052-6399, USA
E-mail: gregmer@microsoft.com

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): D.1.3, C.2.4, F.1.2, D.2.4, I.2.11

ISSN 0302-9743
ISBN 3-540-21044-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN: 10986257 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2949
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Preface

The 6th International Conference on Coordination Models and Languages (Coordination 2004) was held at the Computer Science Department of Pisa University, Italy, on February 24–27 2004. The previous conferences in this series took place in Cesena (Italy), Berlin (Germany), Amsterdam (Netherlands), Limasol (Cyprus), and York (UK). Building on the success of these events, the latest conference provided a forum for the growing community of researchers interested in models, languages, and implementation techniques for coordination and component-based software, as well as applications that exploit them.

The need for increasing programming productivity and rapid development of complex systems provides the pragmatic motivation for the development of coordination/orchestration languages and models. The intellectual excitement associated with such endeavors is rooted in the decades-old desire to leverage off increasingly higher levels of abstractions. Coordination-based methods provide a clean separation between individual software components and their interactions within their overall software organization. Coordination is relevant in design, development, debugging, maintenance, and reuse of all complex concurrent and distributed systems. Specifically, coordination becomes paramount in the context of open systems, systems with mobile entities, and dynamically reconfigurable evolving systems. Moreover, coordination models and languages focus on such key issues in component-based software engineering as specification, interaction, and dynamic compositions. More recently, market trends brought on by the commercialization of the World-Wide Web, have fuelled a new level of interest in coordination-based approaches in industry. Applications like BizTalk, standards like the Web services' WS-* family, and contending coordination standards like BEPL4WS and WSCI, are all examples of this phenomenon. This interest is opening up new opportunities both to apply coordination-based techniques to a broad class of applications as well as to grapple with potentially new kinds of requirements coming from Internet-scale scenarios.

The main topics of the conference included: theoretical models and foundations for coordination, coordination middleware, specification, refinement, and analysis of software architectures, architectural and interface definition languages, agent-oriented languages and models, dynamic software architectures, component programming, Web services, coordination in peer-to-peer and grid computing, tools and environments for the development of coordinated applications, industrial relevance of coordination and software architectures, domain-specific software coordination models, and case studies.

The Program Committee, consisting of 23 members, considered 72 papers and selected 20 for presentation. These papers were selected on the basis of originality, quality and relevance to the topic of the conference. These proceedings include the revised version of the 20 accepted papers, and the abstracts of the invited talks by Gérard Boudol (*Inria Sophia Antipolis, France*), Fabio Casati (*HP Labs, USA*) and Paola Inverardi (*Università dell'Aquila, Italy*).

Paper selection was a difficult and challenging task, and many good submissions had to be rejected. Each submission was refereed by at least four reviewers, and some had five reports or more. We are very grateful to all the program committee members, who devoted much effort and valuable time to read and select the papers. In addition, we gratefully acknowledge the help of a large number of colleagues who reviewed submissions in their area of expertise. They are all listed below. We apologize for any inadvertent omissions.

Following the example of previous editions, we encouraged authors to submit their contributions in electronic format. We handled the submissions with ConfMan (http://confman.unik.no/~confman) a free Web-based paper submission and reviewing system. With reference to this, we would like to thank Antonio Cisternino who managed a lot at crucial steps of the whole process. His computer skill and the time and effort he invested were crucial ingredients in our running of the program committee. Moreover, we would like to thank Roberto Bruni and Emilio Tuosto who managed the activity of editing the proceedings.

Finally, we would like to thank all the authors who submitted their papers for making this conference possible, the Program Committee members, as well as all the conference participants.

February 2003

Rocco De Nicola
Gianluigi Ferrari
Greg Meredith

Program Co-chair

Rocco De Nicola	Università di Firenze, Italy
Greg Meredith	Microsoft, USA

Program Committee

Roberto Amadio	Univ. Marseilles, France
Farhad Arbab	CWI, The Netherlands
Marcelo Bonsangue	Leiden University, The Netherlands
Paolo Ciancarini	Univ. Bologna, Italy
Gianluigi Ferrari	Univ. Pisa, Italy
José Fiadeiro	Univ. Leicester, UK
Chris Hankin	Imperial College, UK
Jean-Marie Jacquet	Univ. Namur, Belgium
Antonia Lopes	Univ. Lisbon, Portugual
Jeff Magee	Imperial College, UK
George Papadopoulos	Univ. Cyprus, Cyprus
Gian Pietro Picco	Politecnico di Milano, Italy
Rosario Pugliese	Univ. Firenze, Italy
Gruia-Catalin Roman	Washington University in St. Louis, USA
Ant Rowstrom	Microsoft Cambridge, UK
Vijay Saraswat	IBM Research, USA
Carolin Talcott	SRI, USA
Robert Tolksdorf	Free University of Berlin, Germany
Herbert Wiklicky	Imperial College, UK
Alan Wood	Univ. York, UK
Franco Zambonelli	Univ. Modena, Italy

Sponsoring Organizations

Università di Pisa

Dipartimento di Informatica
Università di Pisa

Polo Didattico *Fibonacci*
Università di Pisa

Dipartimento di Sistemi e Informatica
Università di Firenze

Microsoft Research
Microsoft Corporation USA

Organizing Chair

Gianluigi Ferrari	Università di Pisa, Italy

Organizing Committee

Andrea Bracciali, Roberto Bruni, Antonio Cisternino, Dan Hirsch, Laura Semini, Emilio Tuosto

Referees

Marco Aldinucci
Andreas Andreou
Pedro Antunes
Lorenzo Bettini
Michele Boreale
Pietro Braione
Antonio Brogi
Roberto Bruni
Marzia Buscemi
Nathan Caswell
Giorgos Chrysanthou
Silvano Dal Zilio
Vincent Danos
Giorgio Ghelli
Daniele Gorla
Juan Guillen-Scholten
Dan Hirsch
Cosimo Laneve
Alessandro Lapadula
Michele Loreti
Marco Mamei
Luis Moniz
Pavlos Moraitis
Amy L. Murphy
Anna Philippou
Darrell Reimer
Laura Semini
Simone Semprini
Leon van der Torre
Emilio Tuosto
Vincent Vanackere
Vasco Vasconcelos
M.J. Wiering
Gianluigi Zavattaro
Sarah Zennou
Peter Zoeteweij

Lecture Notes in Computer Science

For information about Vols. 1–2834

please contact your bookseller or Springer-Verlag

Vol. 2964: T. Okamoto (Eds.), Topics in Cryptology – CT-RSA 2004. Proceedings, 2004. XI, 387 pages. 2004.

Vol. 2957: P. Langendoerfer, M. Liu, I. Matta, V. Tsaoussidis (Eds.), Wired/Wireless Internet Communications. Proceedings, 2004. XI, 307 pages. 2004.

Vol. 2951: M. Naor (Eds.), Theory of Cryptography. Proceedings, 2004. XI, 523 pages. 2004.

Vol. 2949: R. De Nicola, G. Ferrari, G. Meredith (Eds.), Coordination Models and Languages. Proceedings, 2004. X, 323 pages. 2004.

Vol. 2946: R. Focardi, R. Gorrieri (Eds.), Foundations of Security Analysis and Design II. VII, 267 pages. 2004.

Vol. 2943: J. Chen (Eds.), DNA Computing. X, 225 pages. 2004.

Vol. 2930: F. Winkler, Automated Deduction in Geometry. VII, 231 pages. 2004. (Subseries LNAI).

Vol. 2923: V. Lifschitz, I. Niemelä (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 365 pages. 2004. (Subseries LNAI).

Vol. 2916: C. Palamidessi (Eds.), Logic Programming. Proceedings, 2003. XII, 520 pages. 2003.

Vol. 2914: P.K. Pandya, J. Radhakrishnan (Eds.), FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science. Proceedings, 2003. XIII, 446 pages. 2003.

Vol. 2913: T.M. Pinkston, V.K. Prasanna (Eds.), High Performance Computing - HiPC 2003. Proceedings, 2003. XX, 512 pages. 2003. (Subseries LNAI).

Vol. 2911: T.M.T. Sembok, H.B. Zaman, H. Chen, S.R. Urs, S.H. Myaeng (Eds.), Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access. Proceedings, 2003. XX, 703 pages. 2003.

Vol. 2910: M.E. Orlowska, S. Weerawarana, M.M.P. Papazoglou, J. Yang (Eds.), Service-Oriented Computing - ICSOC 2003. Proceedings, 2003. XIV, 576 pages. 2003.

Vol. 2908: K. Chae, M. Yung (Eds.), Information Security Applications. XII, 506 pages. 2004.

Vol. 2906: T. Ibaraki, N. Katoh, H. Ono (Eds.), Algorithms and Computation. Proceedings, 2003. XVII, 748 pages. 2003.

Vol. 2905: A. Sanfeliu, J. Ruiz-Shulcloper (Eds.), Progress in Pattern Recognition, Speech and Image Analysis. XVII, 693 pages. 2003.

Vol. 2904: T. Johansson, S. Maitra (Eds.), Progress in Cryptology - INDOCRYPT 2003. Proceedings, 2003. XI, 431 pages. 2003.

Vol. 2903: T.D. Gedeon, L.C.C. Fung (Eds.), AI 2003: Advances in Artificial Intelligence. Proceedings, 2003. XVI, 1075 pages. 2003. (Subseries LNAI).

Vol. 2902: F.M. Pires, S.P. Abreu (Eds.), Progress in Artificial Intelligence. Proceedings, 2003. XV, 504 pages. 2003. (Subseries LNAI).

Vol. 2901: F. Bry, N. Henze, J. Ma luszyński (Eds.), Principles and Practice of Semantic Web Reasoning. Proceedings, 2003. X, 209 pages. 2003.

Vol. 2900: M. Bidoit, P.D. Mosses (Eds.), Casl User Manual. XIII, 240 pages. 2004.

Vol. 2899: G. Ventre, R. Canonico (Eds.), Interactive Multimedia on Next Generation Networks. Proceedings, 2003. XIV, 420 pages. 2003.

Vol. 2898: K.G. Paterson (Eds.), Cryptography and Coding. Proceedings, 2003. IX, 385 pages. 2003.

Vol. 2897: O. Balet, G. Subsol, P. Torguet (Eds.), Virtual Storytelling. Proceedings, 2003. XI, 240 pages. 2003.

Vol. 2896: V.A. Saraswat (Eds.), Advances in Computing Science – ASIAN 2003. Proceedings, 2003. VIII, 305 pages. 2003.

Vol. 2895: A. Ohori (Eds.), Programming Languages and Systems. Proceedings, 2003. XIII, 427 pages. 2003.

Vol. 2894: C.S. Laih (Eds.), Advances in Cryptology - ASIACRYPT 2003. Proceedings, 2003. XIII, 543 pages. 2003.

Vol. 2893: J.-B. Stefani, I. Demeure, D. Hagimont (Eds.), Distributed Applications and Interoperable Systems. Proceedings, 2003. XIII, 311 pages. 2003.

Vol. 2892: F. Dau, The Logic System of Concept Graphs with Negation. XI, 213 pages. 2003. (Subseries LNAI).

Vol. 2891: J. Lee, M. Barley (Eds.), Intelligent Agents and Multi-Agent Systems. Proceedings, 2003. X, 215 pages. 2003. (Subseries LNAI).

Vol. 2890: M. Broy, A.V. Zamulin (Eds.), Perspectives of System Informatics. XV, 572 pages. 2003.

Vol. 2889: R. Meersman, Z. Tari (Eds.), On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops. Proceedings, 2003. XIX, 1071 pages. 2003.

Vol. 2888: R. Meersman, Z. Tari, D.C. Schmidt (Eds.), On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. Proceedings, 2003. XXI, 1546 pages. 2003.

Vol. 2887: T. Johansson (Eds.), Fast Software Encryption. IX, 397 pages. 2003.

Vol. 2886: I. Nyström, G. Sanniti di Baja, S. Svensson (Eds.), Discrete Geometry for Computer Imagery. Proceedings, 2003. XII, 556 pages. 2003.

Vol. 2885: J.S. Dong, J. Woodcock (Eds.), Formal Methods and Software Engineering. Proceedings, 2003. XI, 683 pages. 2003.

Vol. 2884: E. Najm, U. Nestmann, P. Stevens (Eds.), Formal Methods for Open Object-Based Distributed Systems. Proceedings, 2003. X, 293 pages. 2003.

Vol. 2883: J. Schaeffer, M. Müller, Y. Björnsson (Eds.), Computers and Games. XI, 431 pages. 2003.

Vol. 2882: D. Veit, Matchmaking in Electronic Markets. XV, 180 pages. 2003. (Subseries LNAI).

Vol. 2881: E. Horlait, T. Magedanz, R.H. Glitho (Eds.), Mobile Agents for Telecommunication Applications. Proceedings, 2003. IX, 297 pages. 2003.

Vol. 2880: H.L. Bodlaender (Eds.), Graph-Theoretic Concepts in Computer Science. XI, 386 pages. 2003.

Vol. 2879: R.E. Ellis, T.M. Peters (Eds.), Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003. Proceedings, 2003. XXXIV, 1003 pages. 2003.

Vol. 2878: R.E. Ellis, T.M. Peters (Eds.), Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003. Proceedings, 2003. XXXIII, 819 pages. 2003.

Vol. 2877: T. Böhme, G. Heyer, H. Unger (Eds.), Innovative Internet Community Systems. VIII, 263 pages. 2003.

Vol. 2876: M. Schroeder, G. Wagner (Eds.), Rules and Rule Markup Languages for the Semantic Web. Proceedings, 2003. VII, 173 pages. 2003.

Vol. 2875: E. Aarts, R. Collier, E.v. Loenen, B.d. Ruyter (Eds.), Ambient Intelligence. Proceedings, 2003. XI, 432 pages. 2003.

Vol. 2874: C. Priami (Eds.), Global Computing. XIX, 255 pages. 2003.

Vol. 2871: N. Zhong, Z.W. Raś, S. Tsumoto, E. Suzuki (Eds.), Foundations of Intelligent Systems. Proceedings, 2003. XV, 697 pages. 2003. (Subseries LNAI).

Vol. 2870: D. Fensel, K.P. Sycara, J. Mylopoulos (Eds.), The Semantic Web - ISWC 2003. Proceedings, 2003. XV, 931 pages. 2003.

Vol. 2869: A. Yazici, C. Şener (Eds.), Computer and Information Sciences - ISCIS 2003. Proceedings, 2003. XIX, 1110 pages. 2003.

Vol. 2868: P. Perner, R. Brause, H.-G. Holzhütter (Eds.), Medical Data Analysis. Proceedings, 2003. VIII, 127 pages. 2003.

Vol. 2866: J. Akiyama, M. Kano (Eds.), Discrete and Computational Geometry. VIII, 285 pages. 2003.

Vol. 2865: S. Pierre, M. Barbeau, E. Kranakis (Eds.), Ad-Hoc, Mobile, and Wireless Networks. Proceedings, 2003. X, 293 pages. 2003.

Vol. 2864: A.K. Dey, A. Schmidt, J.F. McCarthy (Eds.), UbiComp 2003: Ubiquitous Computing. Proceedings, 2003. XVII, 368 pages. 2003.

Vol. 2863: P. Stevens, J. Whittle, G. Booch (Eds.), "UML" 2003 - The Unified Modeling Language. Proceedings, 2003. XIV, 415 pages. 2003.

Vol. 2860: D. Geist, E. Tronci (Eds.), Correct Hardware Design and Verification Methods. Proceedings, 2003. XII, 426 pages. 2003.

Vol. 2859: B. Apolloni, M. Marinaro, R. Tagliaferri (Eds.), Neural Nets. X, 376 pages. 2003.

Vol. 2857: M.A. Nascimento, E.S. de Moura, A.L. Oliveira (Eds.), String Processing and Information Retrieval. Proceedings, 2003. XI, 379 pages. 2003.

Vol. 2856: M. Smirnov (Eds.), Quality of Future Internet Services. IX, 293 pages. 2003.

Vol. 2855: R. Alur, I. Lee (Eds.), Embedded Software. Proceedings, 2003. X, 373 pages. 2003.

Vol. 2854: J. Hoffmann, Utilizing Problem Structure in Planing. XIII, 251 pages. 2003. (Subseries LNAI).

Vol. 2853: M. Jeckle, L.-J. Zhang (Eds.), Web Services - ICWS-Europe 2003. VIII, 227 pages. 2003.

Vol. 2852: F.S. de Boer, M.M. Bonsangue, S. Graf, W.-P. de Roever (Eds.), Formal Methods for Components and Objects. VIII, 509 pages. 2003.

Vol. 2851: C. Boyd, W. Mao (Eds.), Information Security. Proceedings, 2003. XI, 453 pages. 2003.

Vol. 2849: N. García, L. Salgado, J.M. Martínez (Eds.), Visual Content Processing and Representation. Proceedings, 2003. XII, 352 pages. 2003.

Vol. 2848: F.E. Fich (Eds.), Distributed Computing. Proceedings, 2003. X, 367 pages. 2003.

Vol. 2847: R.d. Lemos, T.S. Weber, J.B. Camargo Jr. (Eds.), Dependable Computing. Proceedings, 2003. XIV, 371 pages. 2003.

Vol. 2846: J. Zhou, M. Yung, Y. Han (Eds.), Applied Cryptography and Network Security. Proceedings, 2003. XI, 436 pages. 2003.

Vol. 2845: B. Christianson, B. Crispo, J.A. Malcolm, M. Roe (Eds.), Security Protocols. VIII, 243 pages. 2004.

Vol. 2844: J.A. Jorge, N. Jardim Nunes, J. Falcão e Cunha (Eds.), Interactive Systems. Design, Specification, and Verification. XIII, 429 pages. 2003.

Vol. 2843: G. Grieser, Y. Tanaka, A. Yamamoto (Eds.), Discovery Science. Proceedings, 2003. XII, 504 pages. 2003. (Subseries LNAI).

Vol. 2842: R. Gavaldá, K.P. Jantke, E. Takimoto (Eds.), Algorithmic Learning Theory. Proceedings, 2003. XI, 313 pages. 2003. (Subseries LNAI).

Vol. 2841: C. Blundo, C. Laneve (Eds.), Theoretical Computer Science. Proceedings, 2003. XI, 397 pages. 2003.

Vol. 2840: J. Dongarra, D. Laforenza, S. Orlando (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. Proceedings, 2003. XVIII, 693 pages. 2003.

Vol. 2839: A. Marshall, N. Agoulmine (Eds.), Management of Multimedia Networks and Services. Proceedings, 2003. XIV, 532 pages. 2003.

Vol. 2838: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Knowledge Discovery in Databases: PKDD 2003. Proceedings, 2003. XVI, 508 pages. 2003. (Subseries LNAI).

Vol. 2837: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Machine Learning: ECML 2003. Proceedings, 2003. XVI, 504 pages. 2003. (Subseries LNAI).

Vol. 2836: S. Qing, D. Gollmann, J. Zhou (Eds.), Information and Communications Security. Proceedings, 2003. XI, 416 pages. 2003.

Vol. 2835: T. Horváth, A. Yamamoto (Eds.), Inductive Logic Programming. Proceedings, 2003. X, 401 pages. 2003. (Subseries LNAI).

Table of Contents

Invited Talks

Contributed Papers

A Reactive Programming Model for Global Computing

Gérard Boudol

Inria Sophia Antipolis, France

Abstract

In this talk I introduce a programming model for the mobile code, and more generally for programming in a global computing perspective. I first present some requirements for this style of programming, arising from the features and new observables of the global computing context, and I briefly discuss some of the models and programming languages that have been proposed - Obliq, pi-based and Linda-based models, Ambients. I then present a model based on the ideas of "synchronous" programming, that is based on suspension and preemption primitives associated with locally broadcast events. This programming style, providing a notion of reaction and time-out, looks appropriate to address the unreliable character of accessing resources in a global computing context, and to deal with the various kinds of failures – such as unpredictable delays, transient disconnections, congestion, etc. – that arise in a global network.

The proposed model, called ULM, combines a standard imperative and functional style, such as the one of ML or Scheme, with some construct for "reactive" programming. The model also proposes constructs for programming mobile agents, that move together with their state, made of a control stack and a store. This makes the access to references also potentially suspensive, and this is one of the main novelties of the model. The focus of this work is on giving a precise semantics for a small, yet expressive core language. Some examples of this expressiveness are given in the talk.

R. de Nicola et al. (Eds.): COORDINATION 2004, LNCS 2949, p. 1, 2004.

Open Issues and Opportunities in Web Services Modeling, Development, and Management

Fabio Casati

Hewlett-Packard
1501 Page Mill road, MS 1142
Palo Alto, CA, 94304, USA
fabio.casati@hp.com

Extended Abstract

Despite the great interest and the enormous potential, Web services, and more in general service-oriented architectures (SOAs), are still in their infancy. The speed at which software vendors have released middleware and development tools for Web services is unprecedented, but their level of maturity is still far from their counterparts in conventional middleware. This is only natural and common to any novel technology, and will be corrected over time, as more users adopt this technology and provide feedback on what the actual needs are.

Aside from considerations related to the young age of the technology, the interesting aspect of Web services is that they have characteristics that differ from services in conventional middleware, and that can service development and management tools to a new dimension. For example, services in conventional middleware were often tightly coupled and developed by the same team. This means that it was difficult to combine services in different ways, and that service descriptions were rather poor. Web services are instead loosely-coupled and designed for B2B applications. This means that Web services are independent of each other and are invoked by clients who have no access to the service development team. Since service descriptions are all clients have available to understand the service behavior, they are richer than their counterpart in conventional middleware, going well beyond interface descriptions. This factor, along with the increased standardization and the loose coupling, makes *composition* technologies more applicable. The opportunity of composing Web services and of leveraging richer service descriptions are some of the factors that both create more requirements and enable a greater level of automated support from the middleware and from CASE-like development tools.

Other important aspects are related not so much to Web service technology, but to their intended applicability. For example, Web services are typically used to integrate complex systems and not to perform simple computations. Hence, manageability and ease of deployment become often more important than performance.

In this presentation I will elaborate on these and other key aspects of Web services, emphasizing the opportunities they provide and the unsolved challenges they present from a modeling and management perspective.

R. de Nicola et al. (Eds.): COORDINATION 2004, LNCS 2949, p. 2, 2004.

Compositionality, Coordination and Software Architecture

Paola Inverardi

Dipartimento di Informatica
Universitá dell'Aquila

Abstract

In the last decade the way software is produced has been radically changing. Software component technology has witnessed growing popularity with the advent and diffusion of effective component-based infrastructures like CORBA, .Net and Java Beans. Component Based Software Development allowed integration of heterogeneous and legacy components and their availability in innovative contexts like the Internet. At the same time Component Based Software (CBS) systems, often based on Components Off The Shelf (COTS), exhibit severe integration problems at execution time, due to component coordination and synchronization failures [2, 1].

Thus, on one side, component-based infrastructures made easier the assembling of heterogeneous COTS components by providing mechanisms to solve syntactic and static semantics mismatches among components. On the other side, they did not offer support to solve coordination and synchronization problems. This situation is made worse by insufficient component behavioral specifications that make it difficult to establish correct behavioral properties on component assembly.

There has been in the last years a growing interest in the specification and analysis of component assembly [3]. In this respect, CBS development poses old problems in a new context. From a verification point of view the interest is, as usual, in being able to prove behavioral properties of the assembled system. On the negative side we have that in general component behavior is underspecified or unknown, while on the positive side we have that the way components are put together is already known, since it is forced by the software architecture of the component-based infrastructure. As far as behavioral failure recovery is concerned, on the negative side we have that components can be black-box and thus unmodifiable, however on the positive side we can use the software architecture to regulate and control component interaction.

In our research we have identified three different but related research areas:

1. How components, including COTS, can be provided with a behavioral interface useful to prove behavioral properties at the system level.
2. How components are assembled together and what is the impact of the way components are assembled on the behavioral analysis.

R. de Nicola et al. (Eds.): COORDINATION 2004, LNCS 2949, pp. 3–4, 2004.

3. How an assembled system can be checked and made correct with respect to a certain set of behavioral properties.

In my talk I will analyse these three research areas, trying to put our own contribution in the more general perspective of the current research in these different fields and their relations with research in compositionality and coordination.

Our approach uses a specific software architecture as the logical skeleton to assemble components. The information on the software architecture , i.e., on the way components interact together via connectors, allows us to achieve the automatic enforcement of specific coordination policies on the interaction behavior of the components of an assembled system. In our approach, starting from a suitable (and simple) specification of the expected behavior of the assembled system and of its desired behavioral properties, we are able to automatically synthesize architectural connectors that allow only correct interaction (w.r.t. the behavioral properties) to occurr among components [4, 5, 7, 6, 8].

References

[1] B. Boehm and C. Abts, COTS Integration: Plug and Pray?, IEEE Computer, Vol. 32, Num. 1, 1999.

[2] D. Garlan and R. Allen and J. Ockerbloom, Architectural mismatch: Why reuse is so hard. In *IEEE Software Journal, 1995, vol. 12, num. 6.*

[3] Proceeding Component Based Software Engineering Workshops. *CBSE series*, http://sei.cmu.edu.

[4] Paola Inverardi, Simone Scriboni, Connectors Synthesis for Deadlock-Free Component Based Architectures, *IEEE Proc. 16th ASE 2001.*

[5] Paola Inverardi, Massimo Tivoli, Automatic Synthesis of Deadlock free connectors for COM/DCOM Applications. *ACM Proc. FSE/ESEC 2001, Vienna 2001.*

[6] Paola Inverardi, Massimo Tivoli, Deadlock free Software Architectures for COM/DCOM Applications. *Journal of System and Software, CBSE Special Issue 2003.*

[7] A. Bucchiarone, P. Inverardi, M. Tivoli, Automatic synthesis of coordinators of COTS group-ware applications: an example. *DMC 2003, IEEE proceedings WET-ICE 2003.*

[8] P. Inverardi, M. Tivoli, Software Architecture for Correct Components Assembly. In *Formal Methods for the Design of Computer, Communication and Software Systems: Software Architecture, Springer, LNCS 2804.*