Lecture Notes in Computer Science 2603

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
*Berlin*
*Heidelberg*
*New York*
*Barcelona*
*Hong Kong*
*London*
*Milan*
*Paris*
*Tokyo*

Alessandro Garcia  Carlos Lucena
Franco Zambonelli  Andrea Omicini
Jaelson Castro (Eds.)

# Software Engineering for Large-Scale Multi-Agent Systems

## Research Issues and Practical Applications

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Alessandro Garcia
Carlos Lucena
Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio
Rua Marquês de São Vicente, 225, Ed. Pe. Leonel Franca
22453-900 Rio de Janeiro, Brazil
E-mail: {afgarcia,lucena}@inf.puc-rio.br

Franco Zambonelli
Università di Modena e Reggio Emilia
Dipartimento di Scienze e Metodi dell'Ingegneria
Via Allegri 13, 42100 Reggio Emilia, Italy
E-mail: franco.zambonelli@unimo.it

Andrea Omicini
Università di Bologna, Dipartimento di Elettronica, Informatica e Sistemistica (DEIS)
Via Rasi e Spinelli, 176, 47023 Cesena (FC), Italy
E-mail: aomicini@deis.unibo.it

Jaelson Castro
Universidade Federal de Pernambuco, Centro de Informática
Av. Prof. Luiz Freire S/N, 50732-970 Recife PE, Brazil
E-mail: jbc@cin.ufpe.br

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.2, I.2.11, C.2.4, D.1.3, H.5.3

ISSN 0302-9743
ISBN 3-540-08772-9 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

© Springer-Verlag Berlin Heidelberg 2003
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin GmbH
Printed on acid-free paper SPIN: 10872742 06/3142 5 4 3 2 1 0

# Preface

Over the past few years, Object-Oriented Software Engineering has proven to be a powerful paradigm for supporting the development of high-quality software systems. However, we are now faced with the task of engineering large-scale systems composed of pervasive software components that move across and adapt to non-deterministic and open environments, like the Internet, in order to achieve system goals through the coordination of autonomously distributed specialized services. This need for autonomous and pervasive components has spurred the revitalization of the notions and properties associated with software agents and Multi-Agent Systems (MASs). The MAS concepts are being explored to cover most of the software development lifecycle from conceptual modeling and requirements specifications to architectural definition, design and implementation.

MASs and their underlying theories provide more natural support for properties such as autonomy, mobility, environment heterogeneity, organization, openness, and intelligence. As a consequence, agent-based systems are likely to provide new insights into the complexity of developing and maintaining modern software. However, developing robust large-scale agent-based systems will require new software engineering approaches. There are currently many methods and techniques for working with individual agents or systems built using only few agents. Unfortunately, agent-based software engineering is still in its infancy and existing software engineering approaches are unable to cope with large MASs.

MAS features are now being applied to the development of large industrial systems, where such systems involve hundreds, or perhaps thousands of agents. There is a pressing need for software engineering techniques that allow the peculiarities of these systems to be effectively managed, and for methods to guide the process of MAS development. Without adequate development techniques and methods, such systems will not be sufficiently dependable, robust, trustworthy, and extensible. In addition agent-based systems will be difficult to comprehend, and their components will not likely be reusable.

The complexity associated with large MASs is not straightforward. When a huge number of agents interact over heterogeneous environments, various phenomena occur that are not as easy to explain as when only a few agents are working together. As the multiple software agents become highly collaborative and operate in networked environments, they must be context-aware and deal with environment uncertainty. It makes their coordination and management more difficult and increases the likelihood of the occurrence of exceptional situations, such as security holes, privacy violations, and unexpected global effects. Moreover, as users and software engineers delegate more autonomy to their MASs, and put more trust in their results, new concerns arise in real-life applications. However, many existing agent-oriented solutions are far from ideal; in practice, the systems are often built in an ad hoc manner, are error-prone, not scalable, not dynamic, and not generally applicable to

large-scale environments. Commercial success for MAS applications will require scalable solutions based on software engineering approaches in order to ensure effective deployment and to enable reuse.

Today topics related to MASs appear in a wide variety of conference proceedings and journals. However, the research efforts in terms of software engineering for large MASs tend to be scattered in the literature and do not appear as a coherent research topic worth pursuing for its own importance. The papers selected for this book represent one of the first efforts at compiling lessons learned from the application of MAS notions to realistic large-scale software systems. The research presented in this volume illustrates the broad range of software engineering approaches that are being used to cope with the complexity of such systems and to promote the construction of dependable MASs with reusable components. Further, the power of agent-based software engineering is demonstrated through examples that are representative of real-world applications. They describe experiences and techniques associated with large MASs in a variety of problem domains that include network monitoring, e-commerce, Web services, healthcare, and traffic management. The authors have chosen these particular problem domains because they encompass significant kinds of complex problems faced by software engineers practicing in the MAS domain.

Given the comprehensive selection of case studies and software engineering solutions for MAS applications, this book provides a valuable resource for a vast audience of readers. The intended primary audience for this book includes researchers and practitioners who are interested in the progress of software engineering for MASs (especially large, complex systems), individuals interested in understanding the interplay between agents and objects in software development, and those interested in experimental results from MAS applications. Practicing software engineers involved with particular aspects of MASs may find it interesting to learn about experiences in using software engineering approaches to build real systems. A number of chapters in the book discuss the development of MASs from requirements and architecture specifications to implementation. One key contribution of this volume is the description of fresh approaches to reasoning about complex MASs.

This book brings together a collection of 17 papers addressing a wide range of issues in software engineering for large-scale MASs, reflecting the importance of agent properties in today's software systems. The papers presented describe recent developments on specific issues and practical experience. The research issues addressed consist of: (i) a modeling framework, (ii) integration of agent abstractions with other software engineering abstractions and techniques (such as objects, roles, components, aspects, reflection, and patterns), (iii) innovative approaches for coordination and mobility, and (iv) approaches to meeting quality attributes for large-scale MASs, such as dependability, scalability, reusability, and maintainability. At the end of each chapter, the reader will find a list of interesting references for further reading. The book is organized into six parts that deal with topics related to: (i) Software Engineering Foundations, (ii) Requirements Engineering and Software

Architecture, (iii) Coordination and Mobility, (iv) Reuse, (v) Dependability, and (vi) Empirical Studies and Applications.

**Software Engineering Foundations.** The first part of the book focuses on foundations for the development of large agent-based systems. The paper by Silva et al. presents a conceptual framework for modeling large-scale MASs. It identifies abstractions for agent-based software engineering in the light of classical abstractions of object-oriented software engineering. The framework ontology allows for the characterization of large-scale software systems as organizations of agents, objects, and their common and distinguished abstractions. The paper by Odell et al. examines how the notion of role might be used to design complex agent systems. Its authors use social and organizational systems theory as a source of inspiration.

**Requirements Engineering and Software Architecture.** The second part comprises three papers that propose approaches to define requirements and software architectures for agent-oriented systems, and discuss traceability among the artifacts produced in the early stages of MAS development. The paper by Cysneiros and Yu presents a requirements-engineering methodology based on agent concepts such as autonomy, intentionality, and sociality. The proposed methodology complements and extends the *i** modeling framework. The paper by Castro et al. argues that requirements traceability is essential for the development of complex MASs, although it is not currently supported by any of the existing agent-oriented methodologies. Its authors present a framework that extends the Tropos methodology to encompass requirements traceability. The paper by Silva et al. motivates the use of the Reflective Blackboard pattern in the architectural design stage of large-scale MAS development. The proposed pattern provides, early in the architecture definition, the context in which more detailed decisions related to system-level properties, such as coordination, mobility, security, and fault tolerance, can be made in the later stages of large MAS development.

**Coordination and Mobility.** The papers in the third part provide solutions for dealing with coordination and mobility problems in complex MASs. The paper by Roman et al. extends the notion of declarative specifications and provides the mechanisms needed to access the specified resources despite rapid changes in the environment caused by the mobility of hosts, migration of software agents, and changes in connectivity. The paper by Mamei and Mahan focuses on the problem of engineering the respective movements' coordination of huge numbers of agents. It proposes an approach that takes inspiration from the laws of physics. The authors' idea is to have the movements of agents driven by abstract force fields, generated by the agents themselves and propagated via some infrastructure or by the agents in an ad hoc way. The paper by Gustavsson and Fredriksson argues that large MASs emphasize the notion of coordinated behavior and openness, and hence methodological approaches to such systems would benefit from a holistic and context-sensitive framework, i.e., a framework for information ecosystems. According to the authors, the proposed framework allows explicit addressing of important systemic

properties of large MASs, such as coordination, trustworthiness, adaptation, and robustness.

**Reuse.** The fourth part of the book is composed of two papers that are concerned with reusability issues in MASs. The paper by Holvoet and Steegmans first overviews possible ways of reuse in agent-based systems and then presents the MASORG approach to improving MAS reusability. The proposal is largely inspired by three pillars of today's software engineering practice and research, namely separation of concerns, design patterns, and frameworks. In their paper, Pace et al. discuss a CASE environment, called Smartweaver, for supporting the development of aspect-oriented MASs. Smartweaver promotes the reuse of MASs by capturing and keeping agency properties separated from the basic agent functionalities based on concepts from the aspect-oriented paradigm. The paper by Griss discusses the interplay between software components and agents, and how the use of software agents can improve significantly software reuse.

**Dependability.** The papers in the fifth part focus on dependability issues in the development of complex MASs. The paper by Guessoum et al. presents a role-centric approach to evaluating the criticality of software agents and improving their dependability in terms of reliability and availability. The agent criticality is used to replicate agents in order to maximize their dependability based on available resources. The paper by Huhns et al. describes how agent-oriented software engineering can be used to achieve robust large software systems. It also speculates on the implications of multiagent-based redundancy for software development.

**Empirical Studies and Applications.** The last part is concerned with applications, practical problems, and experience in engineering real-world MASs. The paper by Zambonelli et al. reports about an experiment where the behavior observed in dissipative cellular automata resulted in stable macro-level global structures emerging from local interactions among cells. Since dissipative cellular automata exhibit some characteristics of open multi-agent systems, they argue that similar sorts of macro-level behaviors are likely to emerge in MASs and need to be studied, controlled, and fruitfully exploited. They describe some results of a preliminary set of experiments. In their paper, Sycara et al. identify challenges that confront large-MAS designers and claim that these challenges can be successfully addressed by agent-based software engineering based on their vast experience in developing complex MASs. The paper by Tripathi et al. presents their experiences with the development of a MAS for network monitoring. The major challenges in this system were dealing with security problems, the coordination of a huge number of agents to achieve monitoring functions, and fault recovery. The paper by Alencar et al. proposes a declarative agent-based approach to orchestrate and monitor data consistency and business processing. It also describes the current system implementation and applications in which the proposed approach has been applied.

The motivation for the production of this book was the *1st International Workshop on Software Engineering for Large-Scale Multi-Agent Systems* (SELMAS

2002)[1], organized in association with the 24th International Conference on Software Engineering, held in Orlando, Florida, USA, in May 2002. SELMAS 2002 was our first attempt to put together software engineering practitioners and researchers to discuss the multifaceted issues that emerge when using MASs to engineer complex systems. Later the organizers decided to extend the workshop scope covered by the present book. For that purpose they invited several workshop participants to prepare chapters for this book based on their original position papers, and also extended an invitation to a number of other leading researchers in the area to submit additional chapters. After an extensive reviewing process, involving more than 70 reviewers, we selected the papers that appear in this volume.

We are confident that this book will be very useful to the software engineering community, by providing many original and distinct views on such an important interdisciplinary topic, and by contributing to the development of an understanding and crossfertilization among individuals in this research area. It is only natural that the choice of contributors to this book reflects the personal views of the book editors. We raise this point only to suggest that despite the volume of papers and work on software engineering for large-scale MASs, there are still many interesting challenges to be explored. The contributions that can be found in this book are only the beginning. Our thanks go to all the authors, whose work made this book possible. Many of them also helped during the reviewing process. We would like to thank Alfred Hofmann of Springer-Verlag and Juris Hartmanis, LNCS series editor, for recognizing the importance of publishing this book. In addition, we would like to thank the members of the Program Committee who invested many hours reviewing the submitted papers. Also we acknowledge the support and cooperation from Rosa Pinto and Andréa Castor (CIn-UFPE) and the SoC+Agents and TecComm teams (PUC-Rio) which helped us in the preparation of this volume.

January 2003

Alessandro Garcia
Carlos Lucena
Franco Zambonelli
Andrea Omicini
Jaelson Castro

1 Garcia, A., Lucena, C.: Software Engineering for Large-Scale Multi-Agent Systems - SELMAS 2002 (Workshop Report). ACM Software Engineering Notes, Vol. 27, No. 5, September 2002, pp. 82–88.

2002)[1], organized in association with the 24th International Conference on Software Engineering, held in Orlando, Florida, USA, in May 2002. SELMAS 2002 was our first attempt to put together software engineering practitioners and researchers to discuss the multifaceted issues that emerge when using MASs to engineer complex systems. Later the organizers decided to extend the workshop scope covered by the present book. For that purpose they invited several workshop participants to prepare chapters for this book based on their original position papers, and also extended an invitation to a number of other leading researchers in the area to submit additional chapters. After an extensive reviewing process, involving more than 70 reviewers, we selected the papers that appear in this volume.

We are confident that this book will be very useful to the software engineering community, by providing many original and distinct views on such an important interdisciplinary topic, and by contributing to the development of an understanding and crossfertilization among individuals in this research area. It is only natural that the choice of contributors to this book reflects the personal views of the book editors. We raise this point only to suggest that despite the volume of papers and work on software engineering for large-scale MASs, there are still many interesting challenges to be explored. The contributions that can be found in this book are only the beginning. Our thanks go to all the authors, whose work made this book possible. Many of them also helped during the reviewing process. We would like to thank Alfred Hofmann of Springer-Verlag and Juris Hartmanis, LNCS series editor, for recognizing the importance of publishing this book. In addition, we would like to thank the members of the Program Committee who invested many hours reviewing the submitted papers. Also we acknowledge the support and cooperation from Rosa Pinto and Andréa Castor (CIn-UFPE) and the SoC+Agents and TecComm teams (PUC-Rio) which helped us in the preparation of this volume.

January 2003

Alessandro Garcia
Carlos Lucena
Franco Zambonelli
Andrea Omicini
Jaelson Castro

1 Garcia, A., Lucena, C.: Software Engineering for Large-Scale Multi-Agent Systems - SELMAS 2002 (Workshop Report). ACM Software Engineering Notes, Vol. 27, No. 5, September 2002, pp. 82–88.

## Evaluation and Program Committee

P. Alencar (University of Waterloo, Canada)
B. Bauer (University of Augsburg, Germany)
F. Bergenti (University of Parma, Italy)
J.-P. Briot (Laboratoire d'Informatique de Paris 6, France)
C. Bryce (University of Geneva, Switzerland)
J. Castro (UFPE, Brazil)
R. Cerqueira (PUC-Rio, Brazil)
C. Chavez (UFBA, Brazil)
J. Coplien (Bell Laboratories, USA)
D. Cowan (University of Waterloo, Canada)
J. Debenham (University of Technology, Australia)
R. de Lemos (University of Kent, UK)
S. DeLoach (Kansas State University, USA)
M. d'Inverno (University of Westminster, UK)
M. Endler (PUC-Rio, Brazil)
M. Fayad (University of Nebraska, USA)
M. Fontoura (IBM Almaden Research Center, USA)
M. Fredriksson (Blekinge Inst. of Technology, Sweden)
A. Garcia (PUC-Rio, Brazil)
M.-P. Gervais (Laboratoire d'Informatique de Paris 6, France)
P. Giorgini (University of Trento, Italy)
M. Gleizes (University Paul Sabatier, France)
M. Griss (University of California - Santa Cruz, USA)
B. Henderson-Sellers (Univ. of Technology, Australia)
M. Huhns (University of South Carolina, USA)
E. Husita (UEM, Brazil)
C. Iglesias (Politechnical Univ. of Madrid, Spain)
N. Jennings (Southampton University, UK)
C. Jonker (Vrije Universiteit Amsterdam, The Netherlands)
L. Kendall (Monash University, Australia)
R. Kessler (University of Utah, USA)
D. Kinny (University of Melbourne, Australia)
Y. Labrou (Fujitsu Laboratories of America, USA)
J. Leite (PUC-Rio, Brazil)
J. Lind (DFKI, Germany)
C. Lucena (PUC-Rio, Brazil)
J. Maldonado (USP - S. Carlos, Brazil)
M. Mamei (Univ. of Modena & Reggio Emilia, Italy)
D. Marinescu (Central Florida University, USA)
R. Milidiú (PUC-Rio, Brazil)
J. Odell (James Odell Associates, USA)
A. Omicini (University of Bologna, Italy)
V. Parunak (Altarum Institute, USA)
A. Perini (University of Trento, Italy)
A. Rashid (Lancaster University, UK)
N. Rodriguez (PUC - Rio, Brazil)
A. Romanovsky (University of Newcastle, UK)
C. Rubira (UNICAMP, Brazil)
B. Schulze (LNCC, Brazil)
M. Stal (Siemens, Germany)
S. Stobart (University of Sunderland, UK)
K. Sycara (Carnegie Mellon University, USA)
A. Tripathi (University of Minnesota, USA)
G. Wagner (Eindhoven Univ. of Technology, The Netherlands)
G. Weiss (Technical Univ. of Munich, Germany)

# Table of Contents

## Dependability



## Empirical Studies and Applications