

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

2222

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Tokyo

Michael J. Wooldridge Gerhard Weiß
Paolo Ciancarini (Eds.)

Agent-Oriented Software Engineering II

Second International Workshop, AOSE 2001
Montreal, Canada, May 29, 2001
Revised Papers and Invited Contributions



Springer

Series Editors

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

Volume Editors

Michael J. Wooldridge
The University of Liverpool. Department of Computer Science
Chadwick Building, Peach Street, Liverpool L69 7ZF, UK
E-mail: M.J.Wooldridge@csc.liv.ac.uk

Gerhard Weiß
Technische Universität München, Institut für Informatik
80290 München, Germany
E-mail: weissg@informatik.tu-muenchen.de

Paolo Ciancarini
University of Bologna, Dipartimento di Scienze dell'Informazione
Mura Anteo Zamboni 7, 40127 Bologna, Italy
E-mail: ciancarini@cs.unibo.it

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Agent oriented software engineering II : second international workshop ;
revised paper and invited contributions / AOSE 2001, Montreal, Canada, May
29, 2001. Michael J. Wooldridge ... (ed.). - Berlin ; Heidelberg ; New York ;
Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2222)
ISBN 3-540-43282-5

CR Subject Classification (1998): D.2, I.2.11, F.3, D.1, C.2.4

ISSN 0302-9743

ISBN 3-540-43282-5 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 2002
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN: 10840931 06/3142 5 4 3 2 1 0

Preface

Over the past three decades, software engineers have derived a progressively better understanding of the characteristics of complexity in software. It is now widely recognized that *interaction* is probably the most important single characteristic of complex software. Software architectures that contain many dynamically interacting components, each with their own thread of control, and engaging in complex coordination protocols, are typically orders of magnitude more complex to correctly and efficiently engineer than those that simply compute a function of some input through a single thread of control.

Unfortunately, it turns out that many (if not most) real-world applications have precisely these characteristics. As a consequence, a major research topic in computer science over at least the past two decades has been the development of tools and techniques to model, understand, and implement systems in which interaction is the norm. Indeed, many researchers now believe that in future, computation itself will be understood chiefly as a process of interaction.

Since the 1980s, software agents and multi-agent systems have grown into what is now one of the most active areas of research and development activity in computing generally. There are many reasons for the current intensity of interest, but certainly one of the most important is that the concept of an agent as an autonomous system, capable of interacting with other agents in order to satisfy its design objectives, is a natural one for software designers. Just as we can understand many systems as being composed of essentially passive objects, which have state, and upon which we can perform operations, so we can understand many others as being made up of interacting, semi-autonomous agents.

This recognition has led to the growth of interest in agents as a new paradigm for software engineering. The AOSE 2001 workshop sought to examine the credentials of agent-based approaches as a software engineering paradigm, and to gain an insight into what agent-oriented software engineering will look like. AOSE 2001, building on the success of AOSE 2000 (Lecture Notes in Computer Science, Volume 1957, Springer-Verlag), was held at the Autonomous Agents conference in Montreal, Canada, in May 2001. Some 33 papers were submitted to AOSE 2001, following a call for papers on all aspects of agent oriented software engineering, and particularly the following:

- Methodologies for agent-oriented analysis and design
- Relationship of AOSE to other SE paradigms (e.g., OO)
- UML and agent systems
- Agent-oriented requirements analysis and specification
- Refinement and synthesis techniques for agent-based specifications
- Verification and validation techniques for agent-based systems
- Software development environments and CASE tools for AOSE
- Standard APIs for agent programming
- Formal methods for agent-oriented systems, including specification and verification logics

- Engineering large-scale agent systems
- Experiences with field-tested agent systems
- Best practice in agent-oriented development
- Market and other economic models in agent systems engineering
- Practical coordination and cooperation frameworks for agent systems

The present volume contains revised versions of the 14 papers presented at AOSE 2001, together with 5 invited contributions, by Federico Bergenti et al., Jürgen Lind, Morris Sloman, Wamberto Vasconcelos et al., and Eric Yu. It is structured into five parts, reflecting the main issues that arose at the event. We believe this volume reflects the state of the art in the field very well, and hope it will stimulate further exciting development.

Acknowledgments. Thanks to Alfred Hofmann at Springer for his continued support, and for giving us the wonderful volume number (2222!). Thanks also to Adele Howe, the Autonomous Agents 2001 workshops chair, for her excellent technical support. Last – but not least – we would like to gratefully acknowledge all the contributions to the workshop: by the authors, the participants, and the reviewers.

November 2001

M.J. Wooldridge
G. Weiß
P. Ciancarini

Organizing Committee

Michael Wooldridge (chair)
University of Liverpool, UK
email: `M.J.Wooldridge@csc.liv.ac.uk`

Paolo Ciancarini (co-chair)
University of Bologna, Italy
email: `ciancarini@cs.unibo.it`

Gerhard Weiss (co-chair)
Technical University of Munich, Germany
email: `weissg@in.tum.de`

Program Committee

Bernard Bauer (Germany)	Alex van Lamsweerde (France)
Federico Bergenti (Italy)	Jaeho Lee (Korea)
Stefan Bussmann (Germany)	Jürgen Lind (Germany)
Chris Hankin (UK)	Jörg Müller (Germany)
Michael Huhns (USA)	James Odell (USA)
Carlos Iglesias (Spain)	Andrea Omicini (Italy)
Nicholas Jennings (UK)	Van Parunak (USA)
Liz Kendall (Australia)	Robert Tolksdorf (Germany)
David Kinny (Australia)	Jan Treur (The Netherlands)
Joost Kok (The Netherlands)	Leon Sterling (Australia)
Yannis Labrou (USA)	Franco Zambonelli (Italy)

Table of Contents

Part I: Societies and Organizations

Representing Social Structures in UML	1
<i>H. Van Dyke Parunak, James J. Odell</i>	
Diagnosis of the Dynamics within an Organization by Trace Checking of Behavioural Requirements	17
<i>Catholijn Jonker, Ioan Alfred Letia, Jan Treur</i>	
Agent Societies: Towards Frameworks-Based Design	33
<i>Virginia Dignum, Hans Weigand, Lai Xu</i>	

Part II: Protocols and Interaction Frameworks

Bringing Coherence to Agent Conversations	50
<i>Roberto A. Flores, Robert C. Kremer</i>	
Extended Modeling Languages for Interaction Protocol Design	68
<i>Jean-Luc Koning, Marc-Philippe Huget, Jun Wei, Xu Wang</i>	
A Policy Language for the Management of Distributed Agents	84
<i>Naranker Dulay, Nicodemos Damianou, Emil Lupu, Morris Sloman</i>	

Part III: UML and Agent Systems

UML Class Diagrams Revisited in the Context of Agent-Based Systems ...	101
<i>Bernhard Bauer</i>	
Agent Oriented Analysis Using Message/UML	119
<i>Giovanni Caire, Wim Coulier, Francisco Garijo, Jorge Gomez, Juan Pavon, Francisco Leal, Paulo Chainho, Paul Kearney, Jamie Stark, Richard Evans, Philippe Massonet</i>	
Specifying Agent Interaction Protocols with Standard UML	136
<i>Jürgen Lind</i>	
Agents and the UML: A Unified Notation for Agents and Multi-agent Systems?	148
<i>Bernhard Bauer, Federico Bergenti, Philippe Massonet, James J. Odell</i>	

Part IV: Agent-Oriented Requirements Capture & Specification

Modeling Early Requirements in Tropos:	
A Transformation Based Approach	151
<i>Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, John Mylopoulos</i>	
A Requirement Specification Language for Configuration Dynamics of Multi-agent Systems	169
<i>Mehdi Dastani, Catholijn Jonker, Jan Treur</i>	
Determining When to Use an Agent-Oriented Software Engineering Paradigm	188
<i>Scott A. O'Malley, Scott A. DeLoach</i>	
Agent-Oriented Modelling: Software versus the World	206
<i>Eric Yu</i>	

Part V: Analysis and Design

Expectation-Oriented Analysis and Design	226
<i>Wilfried Brauer, Matthias Nickles, Michael Rovatsos, Gerhard Weiß, Kai F. Lorentzen</i>	
Abstractions and Infrastructures for the Design and Development of Mobile Agent Organizations	245
<i>Franco Zambonelli</i>	
Towards an ADL for Designing Agent-Based Systems	263
<i>Marie-Pierre Gervais, Florin Muscutariu</i>	
Automated Derivation of Complex Agent Architectures from Analysis Specifications	278
<i>Clint H. Sparkman, Scott A. DeLoach, Athie L. Self</i>	
A Lifecycle for Models of Large Multi-agent Systems	297
<i>Wamberto Vasconcelos, David Robertson, Jaume Agustí, Carles Sierra, Michael Wooldridge, Simon Parsons, Christopher Walton, Jordi Sabater</i>	
Author Index	319