Lecture Notes in Artificial Intelligence 1994

Subseries of Lecture Notes in Computer Science
Edited by J. G. Carbonell and J. Siekmann

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

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

Jürgen Lind

Iterative Software Engineering for Multiagent Systems

The MASSIVE Method

Springer

Series Editors

Jaime G. Carbonell,Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Author

Jürgen Lind
iteratec GmbH
Inselkammerstr. 4, 82008 Unterhaching, Germany
E-mail: Juergen.Lind@iteratec.de

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Lind, Jürgen:
Iterative software engineering for multiagent systems : the MASSIVE method / Jürgen Lind. - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001
(Lecture notes in computer science ; 1994 : Lecture notes in artificial intelligence)
ISBN 3-540-42166-1

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

ISBN 3-540-42166-1 Springer-Verlag Berlin Heidelberg New York

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

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign
Printed on acid-free paper SPIN 10782125 06/3142 5 4 3 2 1 0

Foreword

Agent-based techniques are beginning to be used to develop a wide range of commercial and industrial applications. This take up is occurring because the agent-based approach offers a natural and powerful means of conceptualising, designing and building complex, distributed systems. The key conceptual components from which this new approach to software engineering derives its power are: (i) the autonomous components (agents) that can achieve their objectives in flexible ways; (ii) the high-level interactions (e.g., cooperation, coordination and negotiation) in which these agents can engage; and (iii) the organisational structures (e.g., teams, coalitions and various forms of hierarchy) into which the agents can arrange themselves. When taken together, the agents represent the application's basic units of computation, the interactions represent the inter-connections between these units and the organisational structures define the way the components relate to one another.

Although the agent-based approach appears to offer a promising new paradigm for building complex distributed systems, to date, the majority of the agent-based applications that have been developed have been built by researchers who specialise in agent-based computing. However, if agent-based computing is to become anything more than a niche technology practised by the few, then the base of people who can successfully use the approach needs to be broadened. A crucial step in this broadening endeavour is to find mechanisms by which professional software engineers can gain access to the philosophy, the concepts and the methods of agent-based computing without having to immerse themselves in the research community. Perhaps the key mechanism for achieving this is to develop methodologies for agent-oriented software engineering. Such methodologies should assist developers in the analysis, design and development of their application; particularly, they should identify the key steps that are involved, the key models that need to be built at the various steps, and how the different models and stages relate to one another.

Against this background, this book, *The* MASSIVE *Method: Software Engineering for Multiagent Systems*, presents one of the first coherent attempts to develop such a methodology for a broad class of agent-based systems. In particular, it provides a clear introduction to the key issues in the field of agent-oriented software engineering and provides a comprehensive overview

of the state of the art. It then describes and illustrates the application of the MASSIVE methodology to a number of real-world applications. When taken together, these components make the book an important contribution to the fledgling field of agent-oriented software engineering and, as such, essential reading for both researchers and practitioners alike.

August 2000 Nick Jennings

Contents

List of Figures

Our ability to imagine complex applications will always exceed our ability to develop them.

Grady Booch

List of Process Models

Our ability to imagine complex applications will always exceed our ability to develop them.

Grady Booch