Handbook on Agent-Oriented Design Processes

Massimo Cossentino • Vincent Hilaire •
Ambra Molesini • Valeria Seidita
Editors

Handbook on Agent-Oriented Design Processes

Springer

Editors

Massimo Cossentino
ICAR-CNR
Palermo
Italy

Vincent Hilaire
University of Technology
of Belfort Montbéliard
Belfort cedex
France

Ambra Molesini
Alma Mater Studiorum
Università di Bologna
Bologna
Italy

Valeria Seidita
DICGIM
University of Palermo
Palermo
Italy

ISBN 978-3-642-39974-9 ISBN 978-3-642-39975-6 (eBook)
DOI 10.1007/978-3-642-39975-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014932973

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Designing a multi-agent system (MAS) is not an easy task: creating agents, environments, norms, organizations, and make them cooperate in order to solve a collective task is both an art and a science.

MAS have to deal with two important and difficult issues: autonomy and interaction. Autonomy is the ability for an agent to solve local and individual problems using its own memory and ability to decide what to do next in accordance with what kind of information it perceives. Of course, agents do not have free will: everything is programmed. It is generally possible at this individual level to have a perfect control of the agent's behavior and to specify its functioning through standard computer design tools and methods. Interaction, on the other hand, may be defined as the mutual influence agents—and the environment—exert to one another. Interaction is a kind of glue that makes agents behave in a loosely coupled way, coordinating their behavior in order to achieve a task, leaving all the little details of the adjustments of actions to their ability to adapt themselves to the dynamic's minor alterations of the whole system.

Autonomy and interaction lead to something that usually computer scientists don't like: the inability to have a perfect control on the whole system. Unpredictability is not something you like when you design software systems. Another amazing thing that happens when you program MAS is that your program does not bug... Usually, when programming standard computer programs, your program fails until you have what you want, and programmers spend their time in the debugger, understanding what is wrong, and why the program breaks down... But this is not true with MAS. Your program does work nearly all the time, but what you see is not what you expect... So debuggers and breakpoints are not as useful as in standard programming, and the MAS programmer has to rely on other tools to understand what is programmed and why the results do not correspond to what is expected.

Why do MAS show such strange behaviors? The main reason is that, in MAS, programming is done at the agent level while results are observed at the system level. And this is both the marvel and the curse of MAS programming. Taken aside, each agent is carefully crafted and programmed to achieve what is expected. But when a group of agents interact in a more or less coordinated way, new properties arise as the result of those interactions, which are difficult to foresee. This collective behavior can solve a lot of tricky issues that would have required hours and hours of delicate design and programming without MAS, if we had to plan everything ahead. And even though, the system would not adapt to new situations as neatly as a MAS. From interaction and autonomy comes magic! Actually this is not magic. It is called emergence, i.e. the ability to produce new results and solve complex problems as a side effect of all the particular actions that agents perform through their coordination of action with other agents. In a MAS, everything is dynamic by nature. Nothing remains the same, and this is this dynamicity which is both required and feared.

But this fantastic feature of MAS has a terrible drawback: the difficulty to control and constrain the global behavior of the system in order for the system to do what you want. And this is precisely why standard methods and tools cannot cope with the specificity of MAS, and why it has been necessary to propose new ones that are collected in this volume.

This book is an invaluable guide, providing both students and advanced practitioners with a thorough compilation of the main methodologies used in MAS. Contributions come from international authors well renowned in the field, who have dedicated their time and effort to the development and application of MAS. This book is the result of decades of intense work while analyzing, designing, and testing multi-agent systems in both research and application domains.

For the first time, a wide panel of MAS methodologies is presented. They range from the reactive point of view, where a solution is obtained as a side effect of an adaptive process, to the more cognitive approach, where agents have beliefs and goals and are defined as members of organizations. Thus, there are both bottom-up and top-down methodologies, allowing MAS designers to get a practical overview of the many approaches available.

Therefore, it is my pleasure to recommend this book that will stand as a reference which long-term users could consult to improve their knowledge and know-how in developing MAS.

Montpellier, France Jacques Ferber
May 2013

Contents

Introduction 1
Massimo Cossentino, Ambra Molesini, Vincent Hilaire, and Valeria Seidita

The IEEE-FIPA Standard on the Design Process Documentation Template 7
Massimo Cossentino, Vincent Hilaire, Ambra Molesini, and Valeria Seidita

ADELFE 2.0 19
N. Bonjean, W. Mefteh, M.P. Gleizes, C. Maurel, and F. Migeon

The ASPECS Process 65
Massimo Cossentino, Vincent Hilaire, Nicolas Gaud, Stephane Galland, and Abderrafiaa Koukam

ELDAMeth Design Process 115
Giancarlo Fortino, Francesco Rango, and Wilma Russo

The Gaia Methodology Process 141
Luca Cernuzzi, Ambra Molesini, and Andrea Omicini

GORMAS: A Methodological Guideline for Organizational-Oriented Open MAS 173
Sergio Esparcia, Estefanía Argente, Vicente Julián, and Vicente Botti

INGENIAS-Scrum 219
Juan C. González-Moreno, Alma Gómez-Rodríguez, Rubén Fuentes-Fernández, and David Ramos-Valcárcel

The O-MASE Methodology 253
Scott A. DeLoach and Juan C. Garcia-Ojeda

PASSI: Process for Agent Societies Specification and Implementation 287
Massimo Cossentino and Valeria Seidita

ROMAS Methodology 331
Emilia Garcia, Adriana Giret, and Vicente Botti

INGENIAS with the Unified Development Process 371
Alma Gómez-Rodríguez, Rubén Fuentes-Fernández,
Juan C. González-Moreno, and Francisco J. Rodríguez-Martínez

The SODA Methodology: Meta-model and Process Documentation 407
Ambra Molesini and Andrea Omicini

The Tropos Software Engineering Methodology 463
Mirko Morandini, Fabiano Dalpiaz, Cu Duy Nguyen,
and Alberto Siena

The OpenUp Process 491
Massimo Cossentino, Vincent Hilaire, and Valeria Seidita

Index 567