Paul Davidsson
Brian Logan
Keiki Takadama (Eds.)

LNAI 3415

# Multi-Agent and Multi-Agent-Based Simulation

Joint Workshop MABS 2004
New York, NY, USA, July 2004
Revised Selected Papers

Springer

Paul Davidsson Brian Logan
Keiki Takadama (Eds.)

# Multi-Agent and Multi-Agent-Based Simulation

Joint Workshop MABS 2004
New York, NY, USA, July 19, 2004
Revised Selected Papers



Springer

Series Editors

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

Volume Editors

Paul Davidsson
Blekinge Institute of Technology
Department of Systems and Software Engineering
37225 Ronneby, Sweden
E-mail: Paul.Davidsson@bth.se

Brian Logan
University of Nottingham
School of Computer Science and IT
Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, UK
E-mail: bsl@cs.nott.ac.uk

Keiki Takadama
Tokyo Institute of Technology
Interdisciplinary Graduate School of Science and Engineering
Dept. of Computational Intelligence and Systems Science
4259 Nagatsuta-cho, Midori-ku, Yokohama, 226-8502 Japan
E-mail: keiki@dis.titech.ac.jp

Library of Congress Control Number: 2005921644

CR Subject Classification (1998): I.2.11, I.2, I.6, C.2.4, J.4, H.4

ISSN 0302-9743
ISBN 3-540-25262-2 Springer Berlin Heidelberg New York



Springer is a part of Springer Science+Business Media

springeronline.com


Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11404668 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3415

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

# Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3416: M. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005.

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005.

Vol. 3398: D.-K. Baik (Ed.), Systems Modeling and Simulation: Theory and Applications. XIV, 733 pages. 2005.

Vol. 3397: T.G. Kim (Ed.), Artificial Intelligence and Simulation. XV, 711 pages. 2005.

Vol. 3396: R.M. van Eijk, M.-P. Huget, F. Dignum (Eds.), Agent Communication. X, 261 pages. 2005.

Vol. 3374: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems. X, 279 pages. 2005.

Vol. 3369: V.R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.), Law and the Semantic Web. XII, 249 pages. 2005.

Vol. 3366: I. Rahwan, P. Moraitis, C. Reed (Eds.), Argumentation in Multi-Agent Systems. XII, 263 pages. 2005.

Vol. 3359: G. Grieser, Y. Tanaka (Eds.), Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets. XIV, 257 pages. 2005.

Vol. 3346: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 249 pages. 2005.

Vol. 3345: Y. Cai (Ed.), Ambient Intelligence for Scientific Discovery. XII, 311 pages. 2005.

Vol. 3343: C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, T. Barkowsky (Eds.), Spatial Cognition IV. Reasoning, Action, and Interaction. XIII, 519 pages. 2005.

Vol. 3339: G.I. Webb, X. Yu (Eds.), AI 2004: Advances in Artificial Intelligence. XXII, 1272 pages. 2004.

Vol. 3336: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. X, 523 pages. 2004.

Vol. 3327: Y. Shi, W. Xu, Z. Chen (Eds.), Data Mining and Knowledge Management. XIII, 263 pages. 2005.

Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004. XX, 987 pages. 2004.

Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004.

Vol. 3276: D. Nardi, M. Riedmiller, C. Sammut, J. Santos-Victor (Eds.), RoboCup 2004: Robot Soccer World Cup VIII. XVIII, 678 pages. 2005.

Vol. 3275: P. Perner (Ed.), Advances in Data Mining. VIII, 173 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004.

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004.

Vol. 3259: J. Dix, J. Leite (Eds.), Computational Logic in Multi-Agent Systems. XII, 251 pages. 2004.

Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004.

Vol. 3248: K.-Y. Su, J. Tsujii, J.-H. Lee, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2004. XVIII, 817 pages. 2005.

Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004.

Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004.

Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), Advances in Natural Language Processing. XII, 488 pages. 2004.

Vol. 3229: J.J. Alferes, J. Leite (Eds.), Logics in Artificial Intelligence. XIV, 744 pages. 2004.

Vol. 3228: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff (Eds.), Formal Approaches to Agent-Based Systems. VIII, 290 pages. 2004.

Vol. 3215: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LVII, 906 pages. 2004.

Vol. 3214: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LVIII, 1302 pages. 2004.

Vol. 3213: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LVIII, 1280 pages. 2004.

Vol. 3209: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web. IX, 201 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004.

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004.

Vol. 3192: C. Bussler, D. Fensel (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 522 pages. 2004.

Vol. 3191: M. Klusch, S. Ossowski, V. Kashyap, R. Unland (Eds.), Cooperative Information Agents VIII. XI, 303 pages. 2004.

Vol. 3187: G. Lindemann, J. Denzinger, I.J. Timm, R. Unland (Eds.), Multiagent System Technologies. XIII, 341 pages. 2004.

Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning. IX, 241 pages. 2004.

Vol. 3171: A.L.C. Bazzan, S. Labidi (Eds.), Advances in Artificial Intelligence – SBIA 2004. XVII, 548 pages. 2004.

Vol. 3159: U. Visser, Intelligent Information Integration for the Semantic Web. XIV, 150 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advances in Case-Based Reasoning. XIII, 822 pages. 2004.

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004.

Vol. 3127: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.), Conceptual Structures at Work. XI, 403 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004.

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004.

Vol. 3097: D. Basin, M. Rusinowitch (Eds.), Automated Reasoning. XII, 493 pages. 2004.

Vol. 3071: A. Omicini, P. Petta, J. Pitt (Eds.), Engineering Societies in the Agents World. XIII, 409 pages. 2004.

Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004. XXV, 1208 pages. 2004.

Vol. 3068: E. André, L. Dybkjær, W. Minker, P. Heisterkamp (Eds.), Affective Dialogue Systems. XII, 324 pages. 2004.

Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), Programming Multi-Agent Systems. X, 221 pages. 2004.

Vol. 3066: S. Tsumoto, R. Słowiński, J. Komorowski, J.W. Grzymała-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004.

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic in Computer Science. X, 275 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004.

Vol. 3055: H. Christiansen, M.-S. Hacid, T. Andreasen, H.L. Larsen (Eds.), Flexible Query Answering Systems. X, 500 pages. 2004.

Vol. 3048: P. Faratin, D.C. Parkes, J.A. Rodríguez-Aguilar, W.E. Walsh (Eds.), Agent-Mediated Electronic Commerce V. XI, 155 pages. 2004.

Vol. 3040: R. Conejo, M. Urretavizcaya, J.-L. Pérez-de-la-Cruz (Eds.), Current Topics in Artificial Intelligence. XIV, 689 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004.

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004.

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004.

Vol. 3020: D. Polani, B. Browning, A. Bonarini, K. Yoshida (Eds.), RoboCup 2003: Robot Soccer World Cup VII. XVI, 767 pages. 2004.

Vol. 3012: K. Kurumatani, S.-H. Chen, A. Ohuchi (Eds.), Multi-Agents for Mass User Support. X, 217 pages. 2004.

Vol. 3010: K.R. Apt, F. Fages, F. Rossi, P. Szeredi, J. Váncza (Eds.), Recent Advances in Constraints. VIII, 285 pages. 2004.

Vol. 2990: J. Leite, A. Omicini, L. Sterling, P. Torroni (Eds.), Declarative Agent Languages and Technologies. XII, 281 pages. 2004.

Vol. 2980: A. Blackwell, K. Marriott, A. Shimojima (Eds.), Diagrammatic Representation and Inference. XV, 448 pages. 2004.

Vol. 2977: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (Eds.), Engineering Self-Organising Systems. X, 299 pages. 2004.

Vol. 2972: R. Monroy, G. Arroyo-Figueroa, L.E. Sucar, H. Sossa (Eds.), MICAI 2004: Advances in Artificial Intelligence. XVII, 923 pages. 2004.

Vol. 2969: M. Nickles, M. Rovatsos, G. Weiss (Eds.), Agents and Computational Autonomy. X, 275 pages. 2004.

Vol. 2961: P. Eklund (Ed.), Concept Lattices. IX, 411 pages. 2004.

Vol. 2953: K. Konrad, Model Generation for Natural Language Interpretation and Analysis. XIII, 166 pages. 2004.

Vol. 2934: G. Lindemann, D. Moldt, M. Paolucci (Eds.), Regulated Agent-Based Social Systems. X, 301 pages. 2004.

# Preface

This volume presents revised and extended versions of selected papers presented at the Joint Workshop on Multi-Agent and Multi-Agent-Based Simulation, a workshop federated with the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), which was held in New York City, USA, July 19–23, 2004. The workshop was in part a continuation of the International Workshop on Multi-Agent-Based Simulation (MABS) series. Revised versions of papers presented at the four previous MABS workshops have been published as volumes 1534, 1979, 2581, and 2927 in the Lecture Notes in Artificial Intelligence series.

The aim of the workshop was to provide a forum for work in both applications of multi-agent-based simulation and the technical challenges of simulating large multi-agent systems (MAS). There has been considerable recent progress in modelling and analyzing multi-agent systems, and in techniques that apply MAS models to complex real-world systems such as social systems and organizations. Simulation is an increasingly important strand that weaves together this work. In high-risk, high-cost situations, simulations provide critical cost/benefit leverage, and make possible explorations that cannot be carried out in situ:

- Multi-agent approaches to simulating complex systems are key tools in interdisciplinary studies of social systems. Agent-based social simulation (ABSS) research simulates and synthesizes social behavior in order to understand real social systems with properties of self-organization, scalability, robustness, and openness.
- In the MAS community, simulation has been applied to a wide range of MAS research and design problems, from models of complex individual agents employing sophisticated internal mechanisms to models of large-scale societies of relatively simple agents which focus more on the interactions between agents.
- For the simulation community, MAS-based approaches provide a new way of organizing and managing large-scale simulations, e.g., Grid-based simulations, and agent simulation presents a challenging new domain requiring the development of new theory and techniques.

The workshop concerned agent simulation construed broadly, from multi-agent approaches to simulating complex systems, to the simulation of part or all of a multi-agent system and the hard technical issues of multi-agent simulation itself. Contemporary directions in both MABS and MAS research present significant challenges to existing simulation tools and methods, such as concepts and tools for modelling complex social systems and environments; scalability (to thousands or millions of large-grain agents); heterogeneity of simulation components and modelled agents; visualization and steering of simulation behavior;

validation of models and results; human-in-the-loop issues; and more. The workshop provided a forum for social scientists, agent researchers and developers, and simulation researchers to assess the current state of the art in the modelling and simulation of social systems and MAS, identify where existing approaches can be successfully applied, learn about new approaches, and explore future research challenges.

We are very grateful to the workshop participants who engaged enthusiastically in the discussions at the workshop, as well as to the authors' engagement in the second round of review and revision of the papers. We would like to thank Franco Zambonelli, the AAMAS 2004 workshop chair, for having selected the workshop among a large number of high-class proposals. We are also grateful to Nick Jennings and Milind Tambe, the AAMAS 2004 general chairs, for having organized such an excellent conference. Particularly, we would like to express our gratitude to Simon Parsons and Elizabeth Sklar, the AAMAS 2004 local organization chairs, for arranging the infrastructure of the workshop.

Finally, we thank Alfred Hofmann and his team at Springer for giving us the opportunity to continue to disseminate the results of the workshop to a broader audience.

Paul Davidsson, Brian Logan, and Keiki Takadama

# Organization

## Program Committee

Gul Agha (University of Illinois at Urbana-Champaign, USA)
John Anderson (University of Manitoba, Canada)
Robert Axtell (Brookings Institution, USA)
Rafael Bordini (University of Liverpool, UK)
Francois Bousquet (CIRAD/IRRI, Thailand)
Christopher D. Carothers (Rensselaer Polytechnic Institute, USA)
Shu-Heng Chen (National Chengchi University, Taiwan)
Claudio Cioffi-Revilla (George Mason University, USA)
Helder Coelho (University of Lisbon, Portugal)
Paul Cohen (USC Information Sciences Institute, USA)
Rosaria Conte (IP/CNR Rome, Italy)
Nick Collier (PantaRei LLC/Argonne National Lab, USA)
Daniel Corkill (University of Massachusetts, USA)
Nuno David (ISCTE, Lisbon, Portugal)
Bruce Edmonds (Manchester Metropolitan University, UK)
Richard Fujimoto (Georgia Institute of Technology, USA)
Nigel Gilbert (University of Surrey, UK)
Nick Gotts (Macaulay Institute, UK)
David Hales (University of Bologna, Italy)
Matt Hare (University of Zurich, Switzerland)
Rainer Hegselmann (University of Bayreuth, Germany)
Wander Jager (University of Groningen, Netherlands)
Marco Janssen (Indiana University, USA)
Christophe Le Page (CIRAD, France)
Scott Moss (University of Manchester, UK)
Emma Norling (University of Melbourne, Australia)
Michael North (Argonne National Laboratory, USA)
Mario Paolucci (IP/CNR Rome, Italy)
Alexander Pretschner (Technische Universität München, Germany)
Patrick Riley (Carnegie Mellon University, USA)
Juliette Rouchier (GREQAM (CNRS), France)
Keith Sawyer (Washington University in St. Louis, USA)
Matthias Scheutz (University of Notre Dame, USA)
Jaime Sichman (University of Sao Paulo, Brazil)
Liz Sonenberg (University of Melbourne, Australia)
Takao Terano (University of Tsukuba, Japan)
Georgios Theodoropoulos (University of Birmingham, UK)
Klaus Troitzsch (University of Koblenz, Germany)

## Organizing Committee

# Table of Contents

## Simulation of Multi-agent Systems



## Technique and Technology

## Methodology and Modelling



## Social Dynamics



## Applications

# Smooth Scaling Ahead: Progressive MAS Simulation from Single PCs to Grids*

Les Gasser, Kelvin Kakugawa, Brant Chee, and Marc Esteva

Graduate School of Library and Information Science,
University of Illinois at Urbana-Champaign
{gasser, kakugawa, chee, esteva}@uiuc.edu

**Abstract.** The emerging "Computational Grid" infrastructure poses many new opportunities for the developing science of large scale multi-agent simulation. The ability to migrate agent experiments seamlessly from simple, local single-processor development tools to large-scale distributed simulation environments provides valuable new models for experimentation and software engineering: first develop local, flexible prototypes, then as they become more stable progressively deploy and experiment with them at larger scales. Currently this kind of progressive scalability is hard for both practical and theoretical reasons: Practically, most agent platforms are designed for just one environment of operation. Smooth scalability is more than a matter of increasing agent numbers. Smooth scaling requires clear integration and consistent alignment between a variety of MAS system and simulation architectures and differing underlying infrastructures. This paper reports on recent progress with our experimental platform MACE3J, which now simulates MAS models seamlessly across a variety of scales and architecture types, from single PCs, to Single System Image (SSI) multicomputers, to heterogeneous distributed Grid environments.

## 1 Introduction

The emerging "Computational Grid" infrastructure [1, 2] poses many new opportunities for the developing science of large scale multi-agent systems. The ability to migrate agent simulations and software seamlessly from simple, local single-processor development tools to large-scale distributed experimental environments will provide valuable new software engineering models for agent-based systems. These new models will couple incremental development of agent simulations with controlled testing and execution. The issue of controlling test and execution is critical because of the complexity of agents as "software engineering units" [3] and because of the need for robustness as experiments are scaled up: the control gained through deterministic simulation is a key engineering tech-

* Work supported by NSF Grant 0208937 and by Fulbright/MECD postdoctoral scholarship FU2003-0569.

P. Davidsson et al. (Eds.): MABS 2004, LNAI 3415, pp. 1–10, 2005.

nique for validating large-scale distributed multi-agent systems and simulations whose behavior can't be easily captured analytically for reasons of complexity.

Scalability is often approached as a problem of size—number of agents, for example. Size *is* a dominant dimension for some types of large distributed systems. For instance, typical program structures used in large-scale distributed scientific computing—a principal target of the Grid initiatives—employ homogeneous, non-interacting components, such as the typical, regularly-structured "data-parallel" codes found in many data analysis packages, and the assumption that software decision logic is stable while data changes from run to run. In these cases, more processing resources translates directly into the ability to process more data and solve "larger" problems.

For large systems with *dynamic, heterogeneous, interacting* components such as MAS simulations, scalability raises several other issues. First, scalability requires having appropriate programming models and tools for building and/or integrating many heterogeneous agents. These tools must manage distributed execution resources and timelines, enforce full encapsulation of agents, and offer tight control over message-based multi-agent interactions. The most useful and general programming tools meeting these requirements are based on *distributed object models* such as the actor model [4]. However, in their pure form distributed object programming models introduce subtle constraints on agent and system construction and operation, which create some difficulties for developing scalable, controllable heterogeneous agent simulations. We detail many of these below.

Second, making simulations smoothly scalable requires aligning simulation models to a variety of programming infrastructures and to a different variety of execution system architectures so that it is possible to conveniently deploy, manage, and control large collections of heterogeneous simulated agents across the different resource pools. For example, simulations designed and run on individual PCs will not have truly concurrent execution, since any single processor necessarily serializes behaviors (see below). In contrast, the so-called *Single System Image (SSI) cluster* technique binds together a collection of resources to create an abstraction of a distributed system as a unified, single-point-of-access pool of concurrent execution resources or threads. At first, this seems like an almost ideal programming abstraction for concurrent agents, because it hides details of resource allocation and agent deployment. However, there are critical tensions between the monolithic *sequential* programming approach offered by the single PC environment, and the actual affordances of other distributed platforms across which multi-agent simulations must be developed and deployed to achieve large scales, such as the monolithic *concurrent* programming approach offered by SSI, and the *heterogeneous, distributed, concurrent* environment of actual deployed Grids. Grids in existence (e.g., Teragrid [5]) are large and diverse collections of execution resources, with heterogeneous, not regular architecture. Care must be taken to align models based on PC and SSI environments to heterogeneous and fully-distributed environments.

Finally, issues such as infrastructure reliability, functional completeness, and the state of documentation for some kinds of environments including SSI and

Grids can be problematic—Grid services and technologies are a case in point, and this state-of-the-art must be accommodated for realistic agent systems.

In the light of these issues, this paper reports on current developments in our MACE3J experimental platform [6]. The principal innovation reported is the ability to abstract away many of these underlying issues of heterogeneity. This abstraction is accomplished with tools that support seamless deployment of large-grain simulation-based multi-agent systems progressively across a variety of system architectures, ranging from single PCs through SSI clusters to fully distributed Grid environments (in our case based on the Globus Toolkit and the Open Grid Services Initiative) [7, 8]. Below we report on the general structure of the MACE3J system that accomplishes this goal. We next introduce several key dimensions across which multi-agent system simulations need to adapt, and we use these as the basis of a taxonomy of progressively scalable architectures that support MACE3J. We illustrate our progress with reference to several deployed experiments that run across all architectures, and show how making these experiments led to insights about the necessary abstractions.

## 2 MACE Overview and Design Philosophy

MACE3J is a scientifically oriented multi-agent testbed whose design philosophy is driven by three objectives [6]. We state these here because they are strongly impacted by the need for smooth scalability across architectures.

**1. Repeatability and control:** MACE3J should support control and randomized repeatability in simulations, which is useful for both development and experimentation.

**2. Transitionable models:** Agents should be built of components that can be transistioned from simulated implementations or environments to real ones.

**3. Generation of knowledge about behavior and structure:** MACE3J should support instrumentation that gathers and analyzes data generated by agents and system behaviors.

### 2.1 Understanding Simulation

A significant focus of MACE3J is simulation support. We view simulation support as the provision of four interlocking types of facilities. MACE3J provides all four of these.

1. ***Modeling facilities*** capture characteristics of modeled systems such as agents or environments in codes that integrate easily with the activation, coordination, and data-gathering services below. MACE3J generalizes the concept of "agents" to *ActiveObjects*, which are defined in MACE3J with a set of interfaces. The ActiveObjects concept captures core functionality that allows for implementation of many different types of "agents", so we use the term to denote the foundations for a range of typical agent types. MACE3J modeling facilities include reusable components for constructing

ActiveObjects, environments, and experiments, coupled with the ability to flexibly import these components and models from other projects.

2. ***Coordination facilities*** provide coherence and synchronization for the distributed objects that make up the MAS model. This includes a selectable combination of deterministic (simulation-driven), user-driven, environment-driven, and/or probabilistic control of simulation events, which allow simulations to be re-run exactly, while supporting probabilistic control of behavioral and timing aspects of simulations such as message delay and system failure (e.g., failure of message delivery or of execution). In MACE3J the fundamental coordination object is called `ActivationGroup`. `ActivationGroup` holds a timeline and a set of coordination routines that control the overall execution profile of a simulation.
3. ***Activation services*** that provide enactment (computing) resources for the distributed objects that make up the model. (Of course, these activation services assume an underlying infrastructure such as a single processor PC and SSI cluster, or a Grid environment.) This includes flexible control and steering of simulations through active user involvement in changing simulation parameters at run-time (blurring the distinction between simulation and enactment and facilitating agent transitions to application).
4. ***Flexible data gathering, management, analysis, and presentation*** is done through user-defined and system-defined probes and data streams.

As a simulation development environment, a key objective of MACE3J has been seamless transitioning of models across execution environments. This kind of model retargeting is valuable as an implementation technology and as a software engineering approach: start simple, validate, and expand to more sophisticated environments while exploiting new capabilities. The single processor (possibly threaded) PC platform is stable, well understood, and controllable, but limited in resources. Agent systems can be developed and prototyped rapidly on the single-processor PC platform because it is highly controllable and accepts heterogeneous, changing codes. In contrast, the Grid is less well understood, less flexible, and works best with more homogeneous and stable codes because of the overhead of distribution, startup, and coordination and because of the underlying heterogeneity of the Grid resources. Thus, we need a progression of different development environments and the ability to link them together in a rational, exploratory development process.

The aims of the tools that support such a progressive development approach are these:

1. Provide a simple, direct system model and API to enable maximum flexibility in Agent styles and granularity.
2. Minimize work for users by providing facilities for distributing, deploying, and controlling agent models.
3. Make coordination lightweight, by abstracting the simulation coordination to simple message patterns implemented in infrastructure.

4. Exploit features of existing platforms such as Grid toolkit services to provide agent simulation layer services, to the extent possible. Current examples include deployment services, directory services and communication services.

## 3 Managing Uncertainty in Scalable MAS Simulations

Here we introduce and develop two main sources of uncertainty for managing design and development of large-scale MAS simulations, with greatest relevance in situations where scale and complexity are related: concurrency and distribution. Concurrency and distribution are inherent properties of MAS, and they introduce several kinds of uncertainty into MAS behaviors.

Concurrency introduces event-ordering uncertainty because concurrently running agents execute at arbitrary rates relative to each other. For a MAS with interactions, increasing scale can increase uncertainty in the ordering of important (interactive) events.

Distribution across space and/or time[1] introduces two types of uncertainty. ***Decision uncertainty*** occurs when information about the states of remote entities (other agents or environments) that could influence local decisions are inaccessible because those states are distributed. ***Semantic uncertainty*** occurs when distribution causes agents to translate communicated references or objects into local interpretations that may vary by local context (e.g., [10]). These kinds of uncertainty are fundamental to MAS. Design tools and processes that help control and incrementally modulate distribution- and concurrency-induced uncertainty ease the complexity of engineering and simulating multi-agent systems.

One aim of deployed MAS is to be able to operate in the presence of these types of uncertainty. However, verification of this is hard to do. Our approach is to manage these types of uncertainty by building into middleware support for strategies of progressive, incremental relaxation of control over uncertainty, to gain confidence and experience.

**Incremental Management of Event-Ordering Uncertainty:** Concurrent execution of agent programs introduces uncertainty about the ordering of interactive agent events such as communications. Event ordering can significantly impact computation results in general, so this uncertainty can have large effects on system reliability, traceability, verifiability, and understandability. Thus, one approach to managing design complexity and improving confidence in MAS behavior is to first eliminate uncertainty in the ordering of events by making events completely repeatable and deterministic. This strict control can then be progressively released to explore system behavior and build confidence under increasing levels of event-ordering uncertainty. In this way, as a MAS experiment is developed, it can be moved to progressively more complex execution environments with progressively greater degrees of freedom in event ordering due to concurrent execution. MACE3J has two ways for exploiting this progressive approach

[1] Other dimensions of distribution beyond space and time are also introduced in [9].