Software Engineering for Large-Scale Multi-Agent Systems SELMAS 2002: Workshop Report

Alessandro F. Garcia   Carlos J. P. Lucena
Departamento de Informática – Grupo TecComm
Pontifícia Univ. Católica do Rio de Janeiro - Brazil
{afgarcia, lucena}@inf.puc-rio.br

Jaelson Castro
Centro de Informática
Universidade Federal de Pernambuco - Brazil
jbc@cin.ufpe.br

Andrea Omicini
Dipart. di Elettronica, Informatica e Sistemistica
Università di Bologna - Italy
aomicini@deis.unibo.it

Franco Zambonelli
Dipart. di Scienze e Metodi dell'Ingegniria
Università di Modena & Reggio Emilia - Italy
franco.zambonelli@unimo.it

PUC-RioInf.MCC19/02 July, 2002

Abstract

This paper reports on the First International Workshop on *Software Engineering for Large-Scale Multi-Agent Systems* (SELMAS 2002) held in Orlando, Florida, USA, May 19, 2002, as part of the *International Conference on Software Engineering* (ICSE 2002). The main purpose behind this workshop is to share and pool the collective experience of people, both academics and practitioners, who are actively working on software engineering for large-scale multi-agent systems. The call for papers elicited about 20 submissions of which 17 papers were accepted for publication in the workshop proceedings and 12 papers were accepted for presentation. A selected set of workshop papers and invited papers are going to appear in the book *Software Engineering for Large-Scale Multi-Agent Systems* (LNCS, Springer, 2002). The workshop consisted of an opening presentation, several paper presentations that were organized into three distinct sessions, and a panel. During the workshop we informally reviewed ongoing and previous work and debated a number of important issues. The SELMAS'02 Web page, including the papers and the electronic version of this report, can be found at <www.teccomm.les.inf.puc-rio.br/selmas2002>. We begin by presenting an overview of our goals and the workshop structure, and then focus on the workshop's technical program.

Keywords: Multi-agent systems, software engineering, distributed systems.

Resumo

Este artigo é um relatório sobre o 1º. *Workshop* Internacional em Engenharia de Software para Sistemas Multi-Agentes de Larga Escala (SELMAS 2002), realizado em Orlando, Florida, USA, em 19 de maio de 2002, como evento integrante da Conferência Internacional em Engenharia de Software (ICSE 2002). O principal objetivo deste *workshop* é compartilhar e sincronizar a experiência coletiva de pessoas, tanto acadêmicas quanto profissionais, que estão ativamente trabalhando em engenharia de software para sistemas multi-agentes complexos. Com a chamada de artigos, nós recebemos cerca de 20 submissões, das quais 17 foram aceitas para publicação nos anais do *workshop* e 12 foram aceitas para apresentação. Um conjunto selecionado de artigos do *workshop* e artigos convidados irão aparecer no livro *Software Engineering for Large-Scale Multi-Agent Systems* (LNCS, Springer, 2002). O *workshop* consistiu de uma apresentação de abertura, várias apresentações de artigos que foram organizadas em sessões técnicas distintas, e um painel. Durante o *workshop*, nós informalmente revisamos trabalhos passados e atuais, bem como debatemos uma série de questões interessantes. A página na Web do SELMAS'02, incluindo os artigos e a versão eletrônica deste relatório, pode ser encontrada em <www.teccomm.les.inf.puc-rio.br/selmas2002>. Nós começamos este artigo apresentando uma revisão dos nossos objetivos e a estrutura do *workshop*, e então focamos no programa técnico do *workshop*.

Palavras-chave: Sistemas multi-agentes, engenharia de software, sistemas distribuídos.

1. Introduction

The advances in networking technologies are paving the way towards building large-scale multi-agent systems (MAS). In the near future, it is expected that multitudes of autonomous software agents will be deployed in our networks and on the Web to pursue goals on our behalf by communicating, and either cooperating or competing with each other [4, 17]. Although software systems built as organizations of autonomous and intelligent agents promise to provide computer scientists and engineers with the expressive and computational power to tackle levels of complexity never reached before, specific abstractions, methodologies, and tools are required to enable an engineered approach to the construction of such systems [17]. There is an urgent need not only for theoretical foundations making the MAS conceptual setting clear, but also for specific methodologies driving the development of complex MAS, and for powerful and manageable infrastructures making MAS a viable approach for building complex, distributed software systems [5, 17].

In this context, research in multi-agent software engineering has been carried out according two different approaches: (i) agent-based software engineering [11, 21], and (ii) object-oriented software engineering for multi-agent systems [1, 2, 5, 6, 12]. Researchers in the first approach argue persuasively that multi-agent systems often are much more complex than object-oriented systems and hence the traditional object model generally fails to capture the complexity of multi-agent systems. In this approach, agents are a new abstraction that substitutes the object abstraction realizing the agent abstraction as a new software engineering paradigm. As a result, proponents of this approach assert it is necessary to develop new software engineering techniques, methods and tools that are specifically tailored to agents. On the other hand, researchers in the second approach propose the integration of agents into the object-orientation world and, thus, they think of objects and agents as complementary abstractions. As a result, they have centered on using and extending existing techniques and methods from object-oriented software engineering, such as design patterns, frameworks, and modeling languages, to multi-agent software engineering. An important aim of both approaches has been to get more insight into how software engineering techniques and methods may alleviate the complexity associated with MAS development, and how they may lead to large, complex agent-based software that is more dependable and more widely reusable without sacrificing efficiency.

The above considerations motivated the organization of the SELMAS'02 workshop. The workshop was also motivated by the advances in the MAS engineering field and our research context in the SoC + MAS group at PUC-Rio that has investigated software engineering techniques and methodologies for building large-scale MAS [18]. The main goals of this workshop were:

1. To determine the overlap and integration of the two general research approaches for multi-agent software engineering;
2. To understand those issues in the agent technology that hinder or improve the production of large-scale distributed systems; and
3. To provide a comprehensive overview of software engineering techniques that may successfully be applied to deal with the complexity associated with realistic multi-agent software.

Other specific interests of the workshop were to collect experience reports regarding empirical studies and software engineering techniques for large-scale multi-agent systems, to identify best practices for MAS development and to establish a research agenda for those researchers interested in multi-agent software engineering. The workshop brought together researchers interested in pushing the frontier in this important and burgeoning area, and practitioners who have experience with MAS development that can help to guide their research. The workshop consisted of an opening presentation, a panel (at the end of the workshop) and three paper sessions, organized around some of the key themes that emerged from the position papers. The paper sessions were introduced by brief presentations, and continued with general discussion.

2. Workshop Proceedings and Program Committee

The papers were collected in the ICSE workshop proceedings [7], and in the SELMAS'02 Web site. The Program Committee (PC) was composed by the following members:

Alessandro Garcia (PUC-Rio – Brazil) - **Chair**
Alexander Romanovsky (University of Newcastle - UK)
Anand Tripathi (University of Minnesota - USA)
Andrea Omicini (University of Bologna – Italy) - **Chair**
Awais Rashid (Lancaster University – UK)
Carlos Lucena (PUC-Rio – Brazil) - **Chair**

Carlos Iglesias (Polytechnic University of Madrid - Spain)
Cecília Rubira (UNICAMP - Brazil)
Ciaran Bryce (University of Geneve - Switzerland)
Christina Chavez (UFBA - Brazil)
Dan Marinescu (University of Central Florida - USA)
Donald Cowan (University of Waterloo - Canada)
Eric Yu (University of Toronto – Canada)
Franco Zambonelli (Univ. of Reggio Emilia – Italy) - **Chair**
Gerd Wagner (Eindhonven Univ. Technology - The Netherlands)
Gerhard Weiss (Technical University of Munich - Germany)
Jaelson Castro (UFPE – Brazil) - **Chair**
James Odell (OMG - USA)
Jie Xu (University of Durham – UK)
Julio Leite (PUC-Rio - Brazil)
Katia Sycara (Carnegie Mellon University - USA)
Liz Kendall (Monash University - Australia)
Marco Mamei (University of Modena - Italy)
Marcus Fontoura (IBM Almaden Research Center - USA)
Mark d'Inverno (University of Westminster – UK)
Martin Fredriksson (Blekinge Institute of Technology - Sweden)
Mike Wooldridge (University of London - UK)
Mohamed Fayad (University of Nebraska - USA)
Nick Jennings (University of Southampton - UK)
Ruy Milidiú (PUC-Rio - Brazil)
Simon Stobart (University of Sunderland - UK)
Van Parunak (Altarum Institute - USA)

3. Workshop Organization and Structure

The organization was under the responsibility of the organizing chairs Alessandro Garcia, Carlos Lucena, Jaelson Castro (UFPE, Brazil), Franco Zambonelli (University of Modena and Reggio Emilia, Italy) and Andrea Omicini (University of Bologna, Italy) and with the assistance of the PC. One full day was allocated for the workshop (May 19, 2002). There were about 30 participants who contributed largely with position papers, which were reviewed and revised before the workshop. We received about 20 submissions from eight countries:

Argentina, Belgium, Brazil, Canada, France, Italy, Sweden and the U.S.. We selected 17 papers for publication in the workshop proceedings and 12 papers for presentation. Each paper was reviewed by at least four members of the PC or additional reviewers; the final selection was made by the workshop organizers based on the evaluation forms. The presented papers were chosen because they offered different or novel perspectives on the workshop topics, and because they had a high potential for generating issues that would stimulate the discussions. An additional description of the selection process, as well as all the participants' position papers, can be found at the SELMAS Web site.

The meeting provided a forum for the exchange of ideas on case studies and diverse approaches to the development of MAS. In preparation to the workshop, participants were requested to read all other submissions, and asked to prepare a clear position statement and questions that were likely to stimulate discussion. Moreover, each presenter tried to identify open questions that could provide the basis for further research in the coming years. The talks were common to all participants, and provided a sense of thematic unity by addressing different important topics in MAS engineering theory and practice. The quality of the presentations at SELMAS'02 was high, and triggered a highly interesting discussion amongst the participants of the workshop – which we would sincerely like to thank for their active participation and the level of their contributions to the debate. Interactions between the participants were lively and the discussion sessions often ran overtime. During the lunch and coffee breaks, the workshop participants discussed the benefits of future collaborations.

The workshop was structured into the following parts:

- An opening presentation was the starting point and introduction for the morning and the afternoon sessions. The presentation was given by Carlos Lucena and Alessandro Garcia. They reported about the meeting topics and goals, and the workshop organization process (see Section 4).

- Three technical sessions provided the opportunity to present theoretical and practical issues about MAS engineering and concerns in practice. The first session addressed structuring and modeling techniques for MAS development. The second session was dedicated to a discussion about dependability and reusability aspects within the context of MAS development. Some empirical studies and new tools for MAS and their specifics were presented in the third technical session. At the end of each presentation, some time was reserved for discussion. To maximize time for discussion, we appointed a chair for each session to coordinate a combined presentation for each group of papers. We found this format to be very effective for identifying commonalties and relative strengths of the various approaches. The most important topics of each session are briefly summarized in Section 6.

- A final panel discussion concluded the workshop. Three panelists answered questions from the audience and debated with each other. Unfortunately, there was too little time to resolve many open issues. More information about the panel and the topics discussed is given below.

Following the successful first workshop a number of workshop papers have been selected for extension and publication in a forthcoming special LNCS volume. Moreover, we will include some invited papers in this special volume. It also is hoped that it will be possible to hold a second workshop as part of the ICSE 2003.

4. Opening Presentation: Setting the Stage

SELMAS'02 began with a kick-off presentation by Carlos Lucena and Alessandro Garcia. Lucena established a brief overview of common concepts and terminology, and the motivation for research on multi-agent software engineering. He advocated that the increasing popularity of the Internet and related technologies (such as Web services, Web semantics and ontology) has revitalized the investigation of agent technology as a promising paradigm for engineering complex distributed software systems. He also commented that in spite of the expressive interest of the computing industry (such as HP [9], IBM [10] and Microsoft [14]) in agent research, and the increasing number of academic researchers interested in investigating aspects of multi-agent software engineering, there still are few publications about the subject in relevant journals and transactions on software engineering.

In addition, Lucena explained that objects and agents are abstractions that exhibit points of similarity, but the development of agent-based software poses other challenges to software engineering since software agents inherently are more complex entities. In addition, a large-scale MAS needs to satisfy multiple stringent requirements such as dependability, interoperability, scalability, maintainability and reusability. However, many existing agent-oriented solutions are far from ideal; in practice, they are often built in an ad-hoc manner and are error-prone, not generally applicable, not scalable and not dynamic.

Garcia reported that over eighty percent of all submitted papers were related to the third workshop goal (see Section 1). Some papers discussed how the agent abstraction might improve or make more difficult the development of large-scale MAS. Few papers explicitly discussed relationships between object-oriented software engineering and multi-agent software engineering. Garcia also explained the selection process for the LNCS volume. The foils of this opening presentation are available on the SELMAS Web site.

5. Workshop Presentations

As we explained above, 12 papers were accepted for presentation. Unfortunately, one of the speakers was not able to travel to Orlando. Thus, we had actually 11 paper presentations during the workshop. There were seven presentations in the morning and four presentations after the lunch break. Twenty minutes were allocated for each presentation, followed by 15 minutes of discussion. The papers and their authors were as follows, with the names of the actual participants in the workshop underlined. Summaries of these presentations are presented in the following section of this workshop report.

- Otavio Silva, Alessandro Garcia, Carlos Lucena (Computer Science Department – TecComm Group , PUC-Rio, Rio de Janeiro - Brazil): *The Reflective Blackboard Architectural Pattern for Developing Large-Scale Multi-Agent Systems.*
- Marco Mamei (Dipartimento di Ingegneria dell'Informazione - Università di Modena e Reggio Emilia, Modena – Italy), Michael Mahan (Nokia Research Center Agent Technology Group 5, Burlington - USA): *Engineering Mobility in Large Multi Agent Systems: A Case Study in Urban Traffic Management.*
- Rune Gustavsson, Martin Fredriksson (Department of Software Engineering and Computer Science, Blekinge Institute of Technology, Ronneby - Sweden): *Sustainable Information Ecosystems.*
- Eric Yu, Luiz Cysneiros (Department of Computer Science, University of Toronto, Toronto - Canada): *Large-Scale Agent Systems: A World Modeling Perspective.*
- Zahia Guessoum, Jean-Pierre Briot, S. Charpentier, Samir Aknine, Olivier Marin, Pierre Sens (LIP6, Université Pierre et Marie Curie, Paris - France): *Dynamic and Adaptive Replication for Large-Scale Reliable Multi-Agent Systems.*
- Gruia-Catalin Roman, Christine Julien (Department of Computer Science, Washington

University, Washington-USA), Amy Murphy (Department of Computer Science, University of Rochester, Rochester-USA): *A Declarative Approach to Agent-Centered Context-Aware Computing in Ad Hoc Wireless Environments.*

- Tom Holvoet, Elke Steegmans (Department of Computer Science, KULeuven, Leuven - Belgium): *Application-Specific Reuse in Multi-Agent System Development.*
- Maria Silvia Gatti, Monia Spallanzani (Dipartimento di Ingegneria dell'Informazione - Università di Modena e Reggio Emilia, Modena - Italy), Andrea Rolli (Dipartimento di Elettronica Informatica e Sistemistica, Università di Bologna, Bologna – Italy), Franco Zambonelli, Marco Mamei (Dipartimento di Scienze e Metodi dell'Ingegneria, Università di Modena e Reggio Emilia, Modena – Italy): *What Can Cellular Automata Tell Us About the Behavior of Large Multi-Agent Systems.*
- Rodion Podorozhny, Dewayne Perry (UT Advanced Research in Software Engineering (UT ARISE), University of Texas Austin, Texas - USA): *A Multi-Agent Framework for an Architecting Process Environment.*
- Andrés Díaz Pace, Marcelo Campo, Federico Trilnik (ISISTAN Research Institute UNICEN University Campus Universitario, Tandil - Argentina): *Smartweaver: An Agent-Based Tool for Aspect-Oriented Development.*
- Arthur Reyes (Department of Computer Science & Engineering, University of Texas at Arlington, Arlington - USA): *Introducing the Marshland Design Optimization Tool for Mobile Multi-Agent Systems.*

6. The Sessions

As mentioned above, we had three sessions of presentations and discussions. Each of the sessions was organized according to common themes in the position paper.

Session 1: Modeling and Structuring MAS – When and How

Chair: *Prof. Gruia-Catalin Roman (Washington University, USA)*

This session was devoted to the "when" and "how" aspects of MAS modeling and structuring. An MAS and its constituent agents exhibit properties that manifest during the different phases of MAS development; different techniques ("how" aspect) must be used to model and structure properly such properties in the suitable development stages ("when" aspect). In other words, this session centralized the discussions about how and when it is worth to handle MAS features like goals, mobility, adaptation, learning, autonomy, planning, coordination and so forth.

- The first presentation, by Otavio Silva, motivated the use of the *Reflective Blackboard* pattern in the architectural design stage ("when" aspect) of large-scale MAS development. This pattern is the composition of two other well-know architectural patterns: the *Blackboard* pattern and the *Reflection* pattern. He explained that the proposed pattern provides, early in the architectural design stage, the context in which more detailed decisions related to system-level properties (e.g. communication, coordination, mobility, security, and fault tolerance) can be made in late stages of software development. The pattern allows a better separation of concerns, supporting the separate handling of these system-level properties by means of the computational reflection technique ("how" aspect).
- In the second presentation, Marco Mamei proposed the use of Co-Fields, a new implementation model to coordinate the respective movements of a large number of agents in an environment. The model is based on the concept of computational force fields ("how aspect"): distributed data structures providing to the agents an abstraction of the environment in terms of force fields driving agents towards the achievement of specific coordination tasks. He commented that his future work would proceed towards two main dimensions. On the one hand he is currently completing the definition of a light, micro-kernel, event-based infrastructure, suitable as a supporting middleware for pervasive applications and resource limited devices. On the other hand, he is trying to extend the Co-Fields model and to formalize it.
- Rune Gustavsson claimed that a fundamental challenge in engineering of large-scale MAS involves evolutionary aspects of living systems. A purposeful and principled reuse of system components or web of services in complex networked environments requires what we could characterize as a systemic approach to evolution. That is, harnessing enabling technologies, such as network-centric applications, service-oriented architectures and new models of connectivity, into opportunities to create new applications. But also, in the process of engineering such systems, to ensure that they fulfill both functional properties as well as non-functional qualities. He argued that contemporary approaches to service-oriented architectures, e.g. .NET does not provide the proper methodologies and tools for

service conjunction in a scalable manner. Gustavsson explained that service discovery and conjunction by means of open patterns of interaction are the basic tools for sustainable system behavior ("how aspect").

- In his talk, the position of Luiz Cysneiros was that agent abstractions, with concepts of autonomy, sociality, rationality, etc., are suitable not only for software agents but also for modeling and analyzing ("when aspect") the world in which these software agents function. He discussed how an agent-oriented approach to world modeling would capture the complexity of social issues better than traditional software development methodologies ("how aspect"). He illustrated his view with some software engineering challenges in terms of large-scale health care information systems.

Session 2: Dependability and Reusability in Large MAS
Chair: *Dr. Martin Griss (HP Research Laboratory, USA)*

In this session, new techniques for improving the reusability and dependability of large-scale MAS were discussed. One special session was devoted to these topics due to the high number of submissions approaching them (seven submitted papers).

- Zahia Guessoum presented her approach for evaluating the criticality of software agents and improving their dependability in terms of reliability and availability. The proposed approach is based on the concepts of roles and degree of activity. The agent criticality is used to replicate agents in order to maximize their dependability based on available resources. Her proposal has been validated based on a fault-tolerant framework (DarX) and a multi-agent framework (DIMA).
- Gruia-Catalin Roman presented the research work on context-aware computing in his laboratory at the University of Washington. Of interest to his research group is the ease with which resources can be acquired and retained in the presence of mobility. In particular, he proposed to extend the notion of declarative specifications to provide the mechanisms needed to maintain access to the specified resources despite rapid changes in the environment caused by the mobility of hosts, migration of software agents, and changes in connectivity.
- Tom Holvoet structured his presentation in three parts. First, he offered an overview of the possible ways for reuse in MAS. Secondly, he presented his approach (called MASORG –

MAS Organizations), which is largely inspired by three pillars of today's software engineering practice and research, namely OO mechanisms, separation of concerns, and design patterns and frameworks. The presented reuse method has two main goals: to describe MAS in an abstract, application-independent way, and to reuse such abstract MAS through application-specific adoptions.

Session 3: Experiments and Tools for Large MAS

Chair: *Prof. Ruy Milidiú (PUC-Rio, Brazil)*

In this session, projects, tools, experiments, and case studies together with the respective approaches used, their results, as well as lessons learned were presented. The intention was to help software engineering researchers to get a feeling for the critical issues to consider in MAS development. The discussion after this session was continued in the final panel discussion.

- Marco Mamei presented the paper of Maria Gatti et al. He reported about an experiment, where the behavior observed in a class of cellular automata (which they defined as "dissipative") resulted in stable macro-level global structures emerging from local interactions among cells. Since dissipative cellular automata exhibit characteristics of open multi-agent systems, they expect that similar sorts of macro-level behaviors are likely to emerge in MAS and need to be studied, controlled, and possibly fruitfully exploited. He described their experience with a preliminary set of experiments.

- Rodion Podorozhny reported on his experience in extending and generalizing the design of the multi-agent aspect of an existing software process environment (SPE). An agent framework has been used to model several architecture recovery processes in this SPE. He claimed that few SPEs pay any attention to agent specification. In the context of his work, the agents are intelligent, autonomous entities that are experts in some part of the process. The process is represented in a software process language, Little-JIL. Programs in Little-JIL describe the coordination and communication that enable them to perform a process. Agents may be human or automata. In either case, they are assigned work and are required to report back the success or failure of that work when they are done. Four domain engineering methodologies were described through a classroom experiment.

- In his talk, Andrés Díaz Pace discussed an agent-based CASE environment, called Smartweaver, for assisting aspect-oriented development (AOD). The AOD paradigm is now recognized as an important level of software organization. However, the diversity of AOD approaches often requires that developers have an important grasp of experience and programming skills to effectively implement aspectual designs. His research is aimed at providing smart guidance to developers to better take advantage of AOD facilities. A novel facet of the Andrés proposal is that it demonstrates that software agents combined with planning techniques to derive a sequence of activities developers should carry out in order to implement aspect-oriented applications.

- In the last talk, Arthur Reyes introduced MArSHLAnd, a design synthesis tool that targets system-level design optimization problems for mobile MASs. This tool discovers satisfactory MAS designs by evaluating a collection of randomly-generated, candidate MAS designs simulated in parallel. Design search space is implemented via genetic algorithms.

7. The Panel

Following the afternoon break, a panel concluded the workshop. The panelists were Prof. Gruia-Catalin Roman, Dr. Martin Griss and Prof. Rune Gustavsson. Marco Mamei played the moderator role. Each panelist presented his viewpoint regarding research interests on the workshop topics. The issues naturally evolved out of the discussions during earlier sessions as points of disagreement or unresolved problems. Again, the discussions were lively and participants learned a lot from each other. This panel was a high point in the day and involved 90 minutes of lively discussion. Since we realized there was not enough time to identify specific issues, the participants concentrated on broad issues about software engineering for large-scale multi-agent systems. Since there was a lot of agreement in the group, we were able to make good progress in our discussions. The panelists and the workshop participants discussed several topics. Below is a review of the panelists' statements.

Prof. Catalin explained that his research interests are related to software agents as architectural components (agents such as unity of modularity, unity of execution, and unity of mobility). Roman asserted that an important property in MAS is openness and, as a consequence, context-awareness (and its different degrees) is a critical point in the development of large-scale MAS. He explained context-aware computing refers to the explicit ability of a software system to detect and respond to changes in its environment. Most current facilities supporting context-awareness are relatively simple and limited in scope. When the needs of the application must reach beyond the basics (e.g., the application requires access to services available at a remote location), the programmer needs to contend with more complex processes that include discovery and communication. While these extra costs may be acceptable in wired networks where connections persist over extended periods of time, in ad hoc networks the complexity of managing frequent disconnection can significantly increase the programming effort. Yet, mobile systems do need access to a broad range of resources, maybe even more so than distributed application.

Dr. Griss described the work on multi-agent software engineering at the HP research laboratory. He commented that his team has largely focused on the functionality, customizability and robustness of the personal assistant and supporting agents. Focusing on the personal and mobile aspects of agents, and for the moment downplaying the shopping and e-commerce aspects, his group has been building a system called "CoolAgent", which involves a group of personal, team and service agents and assistants. These agents and assistants work together to manage calendars, meeting rooms, and teleconference lines to set up distributed meetings. In addition, Dr. Griss advocated the need for studying reusability in the context of MAS. Component reuse is enhanced when components are loosely coupled, flexible and adaptable because components can more easily be combined into novel, robust combinations. So, agent technology is a way of making very flexible reusable components. To some degree, an agent component adjusts itself to its environment, making it more reusable. Finally, he advocated that there is a need for agent standards dealing with challenges in large-scale multi-agent systems, like robustness, inter-operability, security, and so forth.

Prof. Gustavsson claimed that since current approaches for networked information systems with a societal focus can be characterized as emphasizing the notion of coordinated behavior and openness, what we need to do next should be based on lessons learned by other sciences and experimental studies to understand the real problems with developing multi-agent software systems. In addition, he thinks that coordination should be a first-class citizen in MASs.

8. Workshop Discussions, Lessons Learned and Lines for Future Research

The workshop brought together a diverse group of researchers and practitioners experienced in many areas of software engineering for MAS. This allowed the workshop to identify a number of best practices when building MAS, as well as directing the participants to problems requiring further research. Based on the various position statements, a number of assertions were made during the discussions in the sessions and in the panel. These claims can be followed as guidelines when dealing with research in the workshop topics:

- *Coordination is an issue that is central to MAS engineering research.* Roman and Mamei asserted that there is a need for conceiving new ways to let the agents of a system communicate and coordinate with each other in a powerful and flexible way and to support these new paradigms with innovative middleware and programming models.
- *Support for reusability and evolution of MAS can be eased by raising the level of abstraction.* This assertion was explicitly made by Otavio Silva and colleagues with their notion of architectural patterns. The goal is not only to get a better conceptual grip on the problem, but also to focus on levels where changes and reuse can have greater impact on the overall architecture of an agent-based application. Cysneiros also advocated that his approach raises the level of abstraction at which the change impact and reusability analysis is performed.
- *Coordination and mobility raise new problems to fault-tolerance research.* Guessoum advocated that the intriguing problems of fault tolerance in MAS could be classified into two main categories. A first category focuses especially on the reliability of an agent within a multi-agent system; this approach handles the serious problems of communication, interaction and coordination of agents with the other agents of the system. The second category addresses the difficulties of making reliable a mobile agent, which is more exposed to security problems.
- *Design Patterns for MAS.* Design patterns are important vehicles for constructing high-quality software. Architectural patterns define the basic structure of software architecture and of systems which implement that architecture; design patterns are more problem-oriented than architectural patterns, and are applied in later design stages. In this sense, there is a need for design patterns for MAS that refine generic architectural patterns. When Pace asked Otavio Silva how he intends to provide guidelines for MAS development in later design stages, he mentioned the intention of his research group in investigating a language of design patterns to refine his high-level pattern (architectural level) in subsequent software engineering phases (detailed design and implementation phases).

- *Advanced separation of concerns can help with MAS reusability and maintainability.* The complexity associated with MAS development is that there are many different kinds of complex concerns that come up during its software lifecycle, all of which should be recognized and at least some of which should be separated. MAS development methods should deal properly with important dimensions of concern that turn up during each phase, and guide the relationships among such different concerns. The relationships of MAS concerns are very complex since they are naturally overlapping (e.g. mobility and coordination, autonomy and learning, planning and adaptation, and so forth). As a consequence, separation and the composition of concerns must be achieved in different MAS development stages. Empirical studies should evidence the extent to which MAS concerns span lifecycle phases and interact with concerns arising in other phases. Tom Holvoet discussed that AOP [13], in the sense of obliviousness and quantification [3], is an effective way to promote reusability and maintainability of different MAS concerns.
- *Mechanisms and techniques for improving MAS dependability are intriguing research issues.* This conclusion was obvious from the large number of submissions that emphasized the need for such mechanisms and techniques. The different concerns involved in MAS development (e.g., high degree of autonomy and mobility) makes more difficult the promotion of dependability in MAS
- *There is a need for techniques for assuring reliability in non-deterministic MAS.* Zahia Guessoum presented her proposal for improving fault-tolerance in MAS based on the active replication technique. Her research team has performed some experiments applying this replication-based solution in deterministic MAS and they obtained interesting results [8]. However, she advocated that there is a need for promoting empirical studies to understand better the fault-tolerance problems in non-deterministic MAS and providing solution for these potential problems.
- *Conceptual models can help with providing domain-independent support for MAS research.* It is well known that there is a lack of conceptual models widely used in the multi-agent software engineering field. Every researcher has a different notion of agenthood and MAS. In this sense, the workshop participants felt an urgent need for theoretical foundations making the MAS conceptual setting clear according the different agency viewpoints. When Carlos Lucena asked the panel about the basic agent properties, each panelist agreed that interaction, autonomy and adaptation are the basic properties for

agents as proposed in the OMG green paper [16].

- *What multi-agent software engineering different from object-oriented software engineering?* Although the group did not tackle that question explicitly, some of the issues raised during the discussions were relevant in that context. Arthur Reyes pointed out that agents cannot be developed simply with mechanisms, methods and techniques from object-orientation since the main benefits of MAS comes from the notion of emergent behavior. Yet another issue discussed was that of modeling and design methodologies. While standard methodologies like UML could be adapted to model MAS, it was unclear whether the current state of the UML technology is enough to capture all fundamental notions, and to express them at the most suitable level of abstraction. In this context, it was also stated that misusing abstractions as provided by standard methodologies can point to defects or a lack of expressiveness in these abstractions.
- *The handling of emergent behavior is a special challenge to engineers of realistic multi-agent software.* Marco Mamei argued that since distributed multi-agent systems exhibit autonomy of components and openness, undesirable behaviors are likely to make their appearance as soon as agents begin to populate the Internet. This requires methodologies, and tools, to predict and control emergent behavior in large multi-agent systems, and either enabling the exploitation of emergent behavior as additional design dimensions, or the prevention of undesirable behavior.
- *The term "large-scale" may be interpreted in several ways.* Tom Holvoet has discussed that in a given context, a large-scale MAS can be interpreted as a system composed of thousands of agents. From the Cysneiros viewpoint, a large-scale MAS is a software system that requires multiple stakeholders during its development process. Other viewpoints were presented by Roman and Griss. Roman, for example, argued that a large-scale MAS can be achieved not only with increasing the number of agents, but also with increasing the reachability of the agents into the system. Griss advocated that a large-scale MAS can be viewed as a system composed of a high number of agents or agent types. In this sense, we need to make clearer what a large-scale MAS really is for our next meeting.
- *Existing agent-oriented methodologies need to be applied to the development of large-scale MAS.* Cysneiros argued that despite rapid advances in agent technologies, their adoption in mainstream application areas such as large-scale information systems is still

limited. He advocated that a major reason is the lack of systematic methods to guide the development of agent-oriented systems and their application to complex, distributed software systems.

- *Cooperation between academia and industry is important for research in software engineering for large-scale MAS*. The approaches discussed in the workshop were mostly based on academic research. Rune Gustavsson and Zahia Guessoum argued that to evaluate the real value of the approaches, they need to be tested and applied on real world software engineering tasks. It was further remarked that the cooperation between academia and industry is important, if not essential, for further development of the approaches.

9. Conclusions

The particular focus of this first meeting was on the role of MAS in supporting the development of complex, distributed software systems. Altogether, the workshop was a very big success due to the quality of the submitted papers, the level of participation of the audience and the profile of the panelists. The workshop achieved its goal to provide a forum for interactive discussions on the research issues of software engineering for large-scale multi-agent systems. Especially in Session I and Session II, the topics of the position papers were closely related. In several talks during the one-day workshop, the speakers presented items for a research agenda.

SELMAS'02 was our first attempt to put researchers from software engineering together to discuss the multi-faceted issues that emerge in using MAS to engineer complex, distributed systems. Given the level of the contributions, we are confident that the workshop was useful to the multi-agent software engineering community, by providing many original and heterogeneous views on such an interdisciplinary topic as well as several attempts to put everything together. It is our hope that SELMAS'02 was only the first event of a series, meant to provide the agent community with a forum where novel ideas and results can be shared by crossing the boundaries of the many research and application areas that meet in the agent field.

As SELMAS, other important, related workshops were organized to discuss research and practice on multi-agent software engineering (such as AOIS and AOSE workshops [19, 20]). As is evident from these meetings and this workshop report, work on multi-agent software engineering remains to be done. There are a number of ways to learn more about current work and get involved:

- Visit the workshop Web site for details about ongoing work.
- Read the referenced papers in this report and the position papers from SELMAS for background information.
- Contact any of the organizers and authors of the SELMAS papers for more information.

Finally, a high-quality set of workshop papers and invited papers are going to appear in the book *Software Engineering for Large-Scale Multi-Agent Systems* (LNCS, Springer, 2002). In addition, there will be some invited papers. The SELMAS'03 workshop is being planned for ICSE 2003. We look forward to an excellent program also next year.

Acknowledgements

The organizers would like to thank all those who contributed with submissions to the workshop and the program committee members who invested many hours reviewing such submissions. In addition, we thank the session chairs and panelists for the fine work in coordinating the sessions and promoting an interesting panel, respectively. Finally, once again we would sincerely like to thank the SELMAS'02 participants for their active involvement in the meeting and the level of their contributions to the debate.

References

[1] Aridor, Y., and Lange, D. "Agent Design Patterns: Elements of Agent Application Design". In *Proceedings of ACM Autonomous Agents'98*, MN, USA, 1998.

[2] Bigus, J., J. Bigus. Constructing Intelligent Agents with Java – A Programmer's Guide to Smarter Applications. Wiley, Second Edition, 2001.

[3] Elrad, T., R. Filman, A. Bader. Aspect-Oriented Programming. *Communications of the ACM*, Vol. 44, No. 10, October 2002.

[4] Garcia, A., C. Lucena, D. Cowan. Agents in Object-Oriented Software Engineering. *Software: Practice & Experience*, 2002. (Accepted to appear)

[5] Garcia, A., V. Silva, C. Lucena. An Aspect-Oriented Approach for Developing Multi-Agent Systems. In *Proceedings of the Brazilian Symposium on Software Engineering*, Brazilian Computer Society, 2001.

[6] Garcia, A., V. Silva, C. Chavez, C. Lucena. Engineering Multi-Agent Systems with Aspects and Patterns. *Journal of the Brazilian Computer Society*, Special Issue on Software Engineering and Databases, August 2002.

[7] Garcia, A., C. Lucena, J. Castro, A. Ominicini, F. Zambonelli (Eds.). Proceedings of the 1st Workshop on Software Engineering for Large-Scale Multi-Agent Systems. *International Conference on Software Engineering (ICSE 2002)*, Orlando, USA, May 2002.

[8] Guessoum, Z., et al. Dynamic and Adaptive Replication for Large-Scale Reliable Multi-Agent Systems. In *Proceedings of the SELMAS'02*, May 2002, pp. 26-30.

[9] HP Agent Research, http://www.hpl.hp.com/agents/

[10] IBM Agent Research, http://www.research.ibm.com/iagents/

[11] Jennings, N., K. Sycara, M. Wooldridge. A Roadmap of Agent Research and Development. *International Journal of Autonomous Agents and Multi-Agent Systems* 1(1) 7-38, 1998.

[12] Kendall, E., P. Krishna, C. Pathak, C. Suresh. "A Framework for Agent Systems". In: Implementing Application Frameworks – Object-Oriented Frameworks at Work, M. Fayad et al. (editors), John Wiley & Sons, 1999.

[13] Kiczales, G. et al. Aspect-Oriented Programming. In P*roceedings of the European Conference on Object-Oriented Programming (ECOOP'97),* Finland. Springer-Verlag LNCS 1241. June 1997.

[14] Microsoft Agent Research, http://www.research.microsoft.com/adapt/

[15] Odell, J., H. Parunak, B. Bauer. Representing Agent Interaction Protocols in UML. In: *Agent-Oriented Software Engineering*, Paolo Ciancarini and Michael Wooldridge eds., Springer Berlin, pp. 121–140, 2001.

[16] Object Management Group – Agent Platform Special Interest Group. Agent Technology – Green Paper. Version 1.0, September 2000.

[17] Omicini, A., R. Tolksdorf, F. Zambonelli (Eds.) (2000): Engineering Societies in the Agents World: First International Workshop. *Lecture Notes in Artificial Intelligence,* Springer, Vol. 1972, 2000.

[18] Separation of Concerns and Multi-Agent Systems (SoC+MAS Group). URL: www.teccomm.les.inf.puc-rio.br/SoCAgents

[19] The International Workshop series in Agent-Oriented Information Systems, http://www.aois.org/

[20] The International Workshop series in Agent-Oriented Software Engineering, http://www.csc.liv.ac.uk/~mjw/aose/

[21] Wooldridge, M., N. Jennings. Agent Theories, Architectures, and Language: A Survey. In: *Intelligent Agents: ECAI-94 Workshop on Agent Theories, Architectures, Languages*, M. Wooldridge & N. Jennings (Eds.), Berlin, Springer, pp. 1-39.