Software Engineering for Large-Scale Multi-Agent Systems SELMAS 2004: Workshop Report

Ricardo Choren, Alessandro Garcia, Carlos Lucena
[choren, afgarcia, lucena]@inf.puc-rio.br

Martin Griss
School of Computer Science, Carnegie Mellon West, EUA
griss@andrew.cmu.edu

David Kung
Department of Computer Science and Engineering, University of Texas at Arlington, EUA
kung@cse.uta.edu

Naftaly Minsky
Department of Computer Science, Rutgers University, EUA
minsky@cs.rutgers.edu

Alexander Romanovsky
School of Computing Science, University of New Castle upon Tyne, Reino Unido
Alexander.Romanovsky@newcastle.ac.uk

Jaelson Castro
Centro de Informática, Universidade Federal de Pernambuco, Brasil
jbc@cin.ufpe.br

Rogério de Lemos
Department of Computer Science, University of Kent, Reino Unido
r.delemos@kent.ac.uk

Danny Weyns
Department of Computer Science, K.U. Leuven, Bélgica
Danny.Weyns@cs.kuleuven.ac.be

PUC-RioInf.MCC 29/04 July, 2004

Abstract: This paper is intended to sum up the results of the Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS 2004) held in Edinburgh, Scotland, May 24-25, 2004, as part of the International Conference on Software Engineering (ICSE 2004). The main purpose of this workshop was 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 some 24 submissions, of which 14 papers were accepted for presentation. A set of selected workshop and invited papers are to appear in a Lecture Notes in Computer Science (LNCS, Springer) volume. The workshop consisted of an opening presentation, two keynote talks, four technical sessions of paper presentations, and two panels. During the workshop we informally reviewed ongoing and previous work and debated a number of important issues. The SELMAS 2004 website can be found at <http://www.teccomm.les.inf.puc-rio.br/semas2004>. We begin by presenting an overview of our goals and the workshop structure, and then focus on the workshop technical program.

Keywords: Multi-agent systems, SELMAS, workshop report

Resumo: Este trabalho busca sumarizar os resultados do Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS 2004), que ocorreu em Edimburgo, Escócia, nos dias 24 e 25 de maio de 2004, como parte da International Conference on Software Engineering (ICSE 2004). O principal objetivo deste workshop foi compartilhar as experiências dos participantes, tanto membros da academia quanto da indústria, que trabalham com sistemas multi-agentes de larga escala. A chamada de trabalhos recebeu 24 submissões, das quais 14 foram aceitas para apresentação. Um conjunto de artigos selecionados e alguns artigos convidados deverão fazer parte de um volume da série Lecture Notes in Computer Science (LNCS, Springer). O workshop teve uma apresentação de abertura, duas palestras convidadas, quarto seções técnicas de apresentação de artigos e dois painéis. Durante o workshop, foi feita uma revisão do trabalho existente na area. O website do SELMAS pode ser encontrado em <http://www.teccomm.les.inf.puc-rio.br/semas2004>. Neste artigo, começamos apresentando um sumário dos objetivos e da estrutura do workshop e, depois, detalhamos o programa do workshop.

Palavras-chave: Sistemas multi-agentes, SELMAS, relatório de workshop

1. Introduction

Advances in networking technology in the last few years have turned the agent technology into a promising paradigm to engineer complex distributed software systems. Nowadays, it has been applied to a wide range of application domains, including e-commerce, human-computer interfaces, telecommunications, and concurrent engineering. Since a software agent is an inherently more complex abstraction, the development of multi-agent systems (MAS) poses new challenges to Software Engineering.

There are many techniques for dealing with individual agents or systems built using only few agents. However, the agent technology is now being applied to the development of large open industrial software systems. Without adequate development techniques and methods, such systems will not be sufficiently dependable, trustable and extensible. Thus, they will be difficult to comprehend, and their components will not be reusable.

The complexity associated with MAS in an open setting involves numerous facets and dimensions. When a large set of agents interact over heterogeneous environments, several problems appear. It makes their coordination and management more difficult and it increases the probability of exceptional situations, security holes, unexpected global effects, and so on. Commercial success for open agent-based applications will require software engineering approaches in order to enable effective scalable deployment.

The above considerations motivated the organization of the SELMAS 2004 workshop. SELMAS 2004 intended to be a forum to bring together researchers and practitioners to discuss the current state of the art and the future research directions in software engineering for large-scale open MAS. The specific goals of this workshop were:

1. Determine the challenges of engineering open MAS;
2. Understand those issues in the agent technology that make it difficult and/or improve the production of large open systems, and;
3. Provide a comprehensive overview of existing software engineering techniques that may successfully be applied to deal with the complexity associated with open multi-agent software.

Other particular interests of the workshop were to collect experience reports regarding empirical studies, 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 guide their research. The workshop consisted of an opening presentation, three panels and five 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

SELMAS 2004 had formal proceedings [1], published by the IEEE and made available at the IEEE Digital Library. The proceedings included all the papers that were presented in the workshop. The Program Committee (PC) that conducted the paper selection process was composed of the following members:

Paulo Alencar [University of Waterloo, Canada]
Bernhard Bauer [Universität Augsburg, Germany]
Federico Bergenti [Università degli Studi di Parma, Italy]
M. Brian Blake [Georgetown University, USA]
Lotzi Boloni [University of Central Florida, USA]
Paolo Bresciani [IRST, Italy]
Jean-Pierre Briot [CNRS, France]
Jaelson Castro [UFPE, Brazil]
Ricardo Choren [PUC-Rio, Brazil] – **chair**
Scott Cost [University of Maryland, USA]
John Debenham [U. of Technology at Sydney, Australia]
Prem Devanbu [University of California at Davis, USA]
Christophe Dony [Université Montpellier II, France]
Carlos A. Fernandez [U. Politécnica de Madrid, Spain]
Marcus Fontoura [IBM Almaden Research Center, USA]
Martin Fredriksson [Blekinge Inst of Technology, Sweden]
Alessandro Garcia [PUC-Rio, Brazil] – **chair**
Marie-Pierre Gervais [LIP 6, France]
Joseph Giampapa [Carnegie Mellon University, USA]
Paolo Giorgini [Università degli Studi di Trento, Italy]
Marie-Pierre Gleizes [IRIT, France]
Martin Griss [Carnegie-Mellon West, USA] – **chair**
Zahia Guessoum [LIP 6, France]
Olivier Gutknecht [LIRMM, France]
Tom Holvoet [Katholieke Universiteit Leuven, Belgium]
Michael Huhns [University of South Carolina, USA]
Elisa Huzita [Universidade Estadual de Maringá, Brazil]
Mark d'Inverno [University of Westminster, UK]
Catholijn Jonker [Vrije Universiteit, The Netherlands]
Robert Kessler [University of Utah, USA]
David Kung [University of Texas at Arlington, USA] – **chair**
Rogério de Lemos [University of Kent, UK]
Carlos Lucena [PUC-Rio, Brazil] – **chair**
Roger Mailler [University of Massachusetts, USA]
Marco Mamei [U. di Modena e Reggio Emilia, Italy]
Naftaly Minsky [Rutgers University, USA] – **chair**
Eugénio Oliveira [Universidade do Porto, Portugal]
Andrea Omicini [University of Bologna, Italy]
Andrés Díaz Pace [UNCPBA, Argentina]
Anna Perini [IRST, Italy]
Omer Rana [Cardiff University, UK]
Awais Rashid [Lancaster University, UK]
Alexander Romanovsky [U. Newcastle upon Tyne, UK] – **chair**
Gustavo Rossi [U. Nacional de La Plata, Argentina]
Cecília Rubira [UNICAMP, Brazil]
Brian Henderson-Sellers [U. of Tech. Sydney, Australia]

Arndt von Staa [PUC-Rio, Brazil]
Michael Stal [Siemens, Germany]
Walt Truszkowski [NASA, USA]
Michael Weiss [Carleton University, Canada]
Andrea Zisman [City University, UK]

3. Workshop Organization and Structure

The organization was under the responsibility of the organizing chairs Ricardo Choren, Alessandro Garcia, Carlos Lucena (PUC-Rio), Martin Griss (Carnegie-Mellon West, USA), David Kung (University of Texas at Arlington, USA), Naftaly Minsky (Rutgers University, USA) and Alexander Romanovsky (University of Newcastle upon Tyne, UK), with the assistance of the PC. Two full days were allocated for the workshop (May 24-25, 2004). There were about 45 participants who contributed, largely with position papers, which were reviewed and revised before the workshop. We received some 24 submissions from different countries. We selected 14 papers for presentation in the workshop. Each paper was reviewed by at least three 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.

The meeting provided a forum for the exchange of ideas on case studies and diverse approaches to the development of MAS. The talks were common to all participants, providing a sense of thematic unity by addressing different important topics in MAS engineering theory and practice. The quality of the presentations at SELMAS 2004 was high and triggered a highly interesting discussion between workshop participants – whom we sincerely want 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. Furthermore, workshop participants discussed the benefits of future collaborations during the lunch and coffee breaks.

The workshop was structured into the following parts:

- An opening presentation by Carlos Lucena (PUC-Rio, Brazil) was the starting point and introduction for the morning and the afternoon sessions. He reported on the meeting topics and goals and the workshop organization process (see section 4).
- Two keynote talks (one each day) were invited. Gul Agha (University of Illinois at Urbana-Champaign, USA) presented "Models for Large-Scale Multi-agent Systems". He presented theories of actor systems to provide powerful modeling techniques for open distributed computing. Gruia-Catalin Roman (Washington University, USA) gave a presentation about "An Agent-Centered Perspective on Context-Aware Computing". This discussion covered trends in context awareness for agent development in ad-hoc networks.
- Four technical sessions (two each day) provided the framework to present theoretical and practical issues concerning MAS engineering. The first session addressed the methodologies and formal methods for open MAS. The second session was dedicated to presenting the issues on agent mobility. The third session introduced interesting examples of MAS modeling techniques. The fourth session was about advanced separation of concerns and metrics for MAS. At the end of each presentation, time was reserved for discussion. To maximize time,

we appointed a chair for each session to coordinate the discussions. The most important topics of each session are briefly summarized in section 6.

- Two interesting panels (one each day) addressed important workshop topics. The panelists answered questions from the audience and discussed with each other. Unfortunately, there was too little time to resolve many open issues. More information about the panels and the topics discussed is given below.

Some of the workshop papers have been selected for extension and publication in a forthcoming special LNCS volume. Moreover, we will publish some invited papers in this special volume. It is also hoped that it will be possible to hold a fourth edition of the workshop as part of the ICSE 2005.

4. Opening Presentation: Setting the Stage

SELMAS 2004 began with a kick-off presentation by Carlos Lucena. He established a brief overview and the motivation for the workshop. He also explained the selection process for the LNCS volume. This opening presentation is available at the SELMAS 2004 website.

5. Keynote Talks

For the first time, the SELMAS workshop had a line-up of keynote speakers:

Gul Agha. Director of the Open Systems Laboratory at the University of Illinois at Urbana-Champaign and an Associate Professor in the Department of Computer Science. World leader researcher in concurrent computing in distributed systems.

Gruia-Catalin Roman. Chair of the Department of Computer Science of the Washington University in St. Louis. His current research involves the study of formal models and design methods for mobile computing and the development of techniques for the visualization of distributed computations.

The talk given by Gul Agha was about Models for Large-Scale Multi-agent Systems. He stated that one can follow the evolution of computing by looking at the evolution of programming languages, showing an example of the evolution of algorithms representation, form Leibniz binary representation through Petri-nets, passing by Turing Machines and etc. Then, he showed the Actors Model, which unifies objects and concurrency models. The problem with representations is to deal with equivalence. In an open multi-agent system, how one can be sure when a component (agent) can be safely replaced by other component? He presented some theories on specifying trace-based properties and verification algorithms. The approach he introduced was a smart observer that does an intelligent monitoring of the system. However, he stated that in large scale systems, observing the exact state of agents is not possible and that massive interaction of agents is the primary challenge in designing, implementing, and reasoning about large scale systems. He concluded by stating that there is a need a more expressive model of time to show the notion of distance and distribution, and the space-time cone of causal influence. These will be models for science and they may rely on characteristics of other science models, such as economics and biology.

The talk given by Gruia-Catalin Roman was about Context Awareness. Context-awareness refers to a computing paradigm in which the behavior of individual components (agents) is determined by the circumstances in which they find themselves to an extent that greatly exceeds the system/environment interaction patterns commonly observed in modern computing. It is assumed that the environment has an exceedingly powerful impact on a particular application component either because the latter needs to adapt in response to changing external conditions or because it needs to rely on resources whose availability is subject to continuous change. In the research his group is conducting, they seek to develop a systematic understanding of the quintessential nature of context-aware computing. They start with the basic premise that, in its most extreme form, context should be made manifest in a manner that is highly local in appearance and decoupled in fact. Furthermore, they assume a notion of context that is relative to the needs of each individual component, and we expect context-awareness to be maintained in a totally transparent manner with little or no programming effort. They explore both formal models and middleware constructs that embody these formative assumptions. Throughout their investigation they make every effort to preserve minimal concepts and elegance of notation.

6. Workshop Paper Presentations

As we explained above, 14 papers were accepted for presentation. In each day, there were two technical sessions. Each speaker had 20 minutes per presentation, followed by 5 minutes for discussion. The papers and their authors were as follows. Summaries of these presentations are presented in the following section of this workshop report.

- *Comparing Agent-Oriented Methodologies Using NFR Approach*, by Carla Silva, Patrícia Tedesco, Jaelson Castro and Rosa Pinto
- *Systematic Integration between Requirements and Architecture*, by Lucia Bastos and Jaelson Castro
- *A Generative Approach for Open Multi-Agent Systems*, by Uirá Kulesza, Alessandro Garcia, Carlos Lucena and Paulo Alencar
- *Formal Specification and Verification of Component Roles in Open Multi-agent Systems*, by Nabil Hameurlain
- *On Efficient Communication and Service Agent Discovery in Multi-agent Systems*, by Myeong-Wuk Jang and Gul Agha
- *Context-Sensitive Data Structures Supporting Software Development in Ad Hoc Mobile Settings*, by Jamie Payton, Gruia-Catalin Roman and Christine Julien
- *Context-Sensitive Access Control for Open Mobile Agent Systems*, by Christine Julien, Jamie Payton and Gruia-Catalin Roman
- *CAMLE: A Caste-Centric Agent-Oriented Modeling Language and Environment*, by Lijun Shan and Hong Zhu
- *Agent-Oriented Modeling Using ANote*, by Ricardo Choren and Carlos Lucena
- *Combining Adaptive Behavior and Role Modeling with Statecharts*, by Danny Weyns, Elke Steegmans and Tom Holvoet
- *Modeling Agent-Oriented Information Systems for Business Processes*, by Ricardo Bastos and Marcelo Ribeiro
- *Advanced Separation of Concerns for Mechatronic Multi-Agent Systems through Dynamic Communities*, by Florian Klein and Holger Giese
- *Separation of Concerns in Open Multi-Agent Systems: An Architectural Approach*, by Alessandro Garcia, Uirá Kulesza and Carlos Lucena

- *Towards a Minimal Performance Metrics Suite for Agent-Based Systems*, by Amir Zeid and Maha Abdel Kader

7. Technical Sessions

As mentioned above, there were four sessions of presentations and discussions. Each of the sessions was organized according to common themes in the position papers. The session summaries, as produced by the respective session chair, are presented below.

TS 1: Methodologies and Formal Methods for Open MAS
Chair: Jaelson Castro (UFPE, Brazil)

Technical session consisted of 4 presentations related to agent-oriented methodologies. The first work was presented by Jaelson Castro and addressed the issue of methodologies comparison using a non functional requirements approach. It was emphasized that agent-oriented development methodologies are difficult to evaluate since they usually differ in their premises, covered phases, models, concepts and the supported multi-agent system properties. It was claimed that the initial step is to define a set of agent properties/attributes of interest. It was shown that the NFR (Non-Functional Requirements) framework is appropriate to represent those high level properties, refinement and relationships. Then two well-known agent oriented methods (GAIA and TROPOS) were compared using the approach. It was explained that the catalogue of agent attributes could of assistance to newcomers which may be willing to choose a methodology to work with. Of course the set of attributes can be adapted and refined to better service the task at hand. Care should be taken to avoid certain biases towards a specific methodology. Moreover, methodologies should also be used in real cases studies to assess their limitations. Unfortunately, few reports are available for consultation.

The second talk in the session was presented by Alessandro Garcia, from PUC-Rio, who presented a generative approach which explores the multi-agent systems domain to enable the code generation of heterogeneous agent architectures. The proposed approach deals with orthogonal (non-crosscutting) and crosscutting agent features since early development phases in a uniform way. It was discussed that the work relied on a pre-defined set of agent-specific concerns. It was claimed that some crosscutting concerns (aspects), such as mobility, can also be incorporated with some limited effort, while other, such as security and adaptability may require some further refinement. It is worth mentioning that the approach was aimed at aspect-oriented programming languages that extend existing object-oriented programming languages, since agent-oriented languages are still quite immature. Further industrial case studies are currently underway.

The third presentation, given by Jaelson Castro, addressed the issue of understanding and supporting the interaction among software requirements and architecture within the TROPOS framework. An approach called SIRA was proposed to related organizational goals and policies to system actor, roles, responsibilities, and constraints. The process has three activities, the first one (Analyzing Elements) takes as inputs the i* requirement models to generate the SIRA element named System Group Schema. The second activity is Selecting Architecture. It relies on the use of Non-Functional Requirements framework (NFR) [nfr00] to select architectural styles, based on non-functional requirements extracted from i* models. The last activity, relating elements links the System Group Schema to architectural elements. It generates the architectural models. It is worth noticing that some high level requirements such as security, needs to be properly refined during the late requirements phase. Some case studies, related to the banking domain are currently under development.

The last presentation of the section was on formal specification of roles. It was presented by Nabil Hameurlain, from LIUPPA Laboratory, France. The work identified requirements for modeling role-based interactions. The RICO (Role-Based Interaction Component) model for specifying agent roles was described. It is important to stress that the RICO model focuses in functional properties, disregarding non-functional issues. It was shown how the Rico model can be mapped to a concurrent formal object-oriented language, the CoOperative Object (COO) formalism that enables the formal specification, analysis and validation of open and concurrent systems. Finally, it was presented how to specify and check properties of a role Component exploiting facilities of the SYROCO, and environment supporting COO, and Petri nets theory: structural properties reachability graphs for behavioral analysis and namely other tools such as INA tool for checking temporal properties.

TS 2: Issues on Agent Mobility
Chair: Alexander Romanovsky (U. Newcastle upon Tyne, UK)

The main theme of technical session 2 was agent mobility. This session started with the presentation entitled On Efficient Communication and Service Discovery in Multi-Agent Systems delivered by Myeong-Wuk Jang. This work is performed in the context of the Actor Architecture platform which is under development in University of Illinois.

Two closely-related problems where introduced in the talk: supporting efficient message passing in large MAS when the agents are mobile and facilitating agent discovery in open MAS. The solution to the first problem put forward by the authors is in using dynamic names for the agents. The second problem is addressed by using a specialized broker agent.

The second talk on Context-Sensitive Data Structures Supporting Software Development in Adhoc Mobile Systems was given by Jamie Payton. It started with a comprehensive introduction of the concept of context-sensitive date structures as a novel paradigm to be used in developing open MAS. In the following part of the talk the ways the programming with context-sensitive data structures should be carried on were presented. In the rest of the talk an approach to implementing the infrastructure support were discussed, it includes protocols for tailored contexts, data structure population, and data access and removal.

The last talk of the session was on Context-Sensitive Access Control for Open Mobile Agent Systems. In this presentation Christine Julien introduced a novel access mechanism suitable for a wide-range of MAS with agent mobility that is based on the general coordination model. This mechanism extends the way the data are accessed by adding a number of new features (profiles, parameters, credentials, pattern, owner's profile, etc.) that allow the system to control (and when necessary to restrict) the data access and to impose application-specific access policies in a very flexible fashion.

Then, Christine described how this functionality was integrated into EgoSpaces coordination model and middleware. A design of a simple music sharing application was presented as a case study demonstrating the main ideas of the context-sensitive access control.

TS 3: MAS Modeling
Chair: Danny Weyns (K.U. Leuven, Belgium)

In this session, a number of new modeling languages and tools for the development of multi-agent systems were presented. The session had four presentations.

Lijun Shan presented the agent-oriented modeling language CAMLE. CAMLE defines a modeling process and specification language for the integral engineering process of multi-agent-based information systems. Central to the CAMLE approach is the Caste model that defines an information system as a collection of agents that maintain dynamic relations to one another as members of certain groups, called castes. CAMLE provides a software environment to support the development of applications.

Ricardo Choren reported on the ANote modeling language. ANote offers a set of diagrams to model a multi-agent system in different views. Static views are used to model the system's goals, agents and environment. Dynamic views define the behavioral properties of the system, i.e. scenarios, plans and interactions. Finally, organizational view defines the structure of a multi-agent system. In ANote, an organization consists of a set of agents that offer services through an interface. Organizations can participate in dependency relations. A dependency relation expresses that an agent of one organization requires a service of another agent in a second organization. The speaker illustrated ANote with a case study on an insurance brokering system.

In her talk, Elke Steegmans demonstrated how adaptive behavior of agents can be combined with statechart-based role modeling. Agents must be adaptive in order to exhibit suitable behavior in changing circumstances of the system. Therefore, a free-flow architecture for agent behavior is proposed. Traditional free-flow architectures however, do not support any structuring, resulting in complex models for non-trivial agents. In the talk, an approach to structure a free-flow architecture was proposed based on roles and interdepecincies between roles. A state chart modeling language is used to guide the software engineer with the design of adaptive agent behavior with free-flow architectures.

On behalf of the authors, Prof. Rafael Prikladnicki discussed the modeling of agent-oriented information systems for business processes. The proposed modeling approach is an extension of the Rational Unified Process. The models are specified through successive refinements, using use cases as a reference to express the system requirements. The presented models are basically an extension of AUML diagrams and notations.

TS 4: Advanced Separation of Concerns and Metrics for MAS
Chair: Martin Griss (Carnegie-Mellon West, USA)

The last technical session was about Advanced Separation of Concerns and Metrics for MAS. The session had three presentations.

Florian Klein presented an advanced separation of concerns for mechatronic multi-agent systems through dynamic communities. The work intended to draw purposeful behavior emerges from seemingly chaotic interaction. He argued that is difficult to design and understand emergent behavior purely analytically. The paper proposed two models: Physical Domain Model and Conceptual Model. The Physical Domain Model models the ontology of the system along with the processes, actuators and sensors. The Conceptual Model specifies the social structure, the roles and patterns, the behavioral patterns and the professed intentions of agents.

Alessandro Garcia demonstrated an architectural approach for the separation of concerns in open MASs. Multiple agent types (e.g. reactive, cognitive) require heterogeneous architectures. Thus, there is a need for more flexible architectural approaches. The proposed aspect-oriented architecture requires a configuration (definitions) in five steps: agent kernel, agent-hood properties, agent types, additional properties and additional properties. Then, an aspect-oriented pattern language is used to implement the multi-agent system, which supports a smooth transition from the architecture to the implementation.

Amir Zeid reported on a performance metrics suite for agent-based systems. The study presented the research method used to derive a minimal set of agent-oriented metrics. The findings of this study indicated that there should be considered four new metrics for multi-agent systems: mobility performance, communication performance, export agent coupling and export agent coupling. Zeid concluded that this work is at its beginning and that much must be done to come up with a set of metrics for multi-agent systems.

8. Panels

During the workshop, two panels were organized to discuss important topics of MAS engineering. The panelists answered questions from the audience and discussed with each other. On the first day, there was the panel on agent dependability in open architectures and on the second day, there was the panel on agent-oriented methodologies and modeling languages. The panel summaries, produced by their respective chairs, are presented below.

P 1: Agent Dependability in Open Architectures
Chair: Rogério de Lemos (University of Kent, UK)
Panelists: Gul Agha, Holger Giese and Gruia-Catalin Roman

The moderator, Rogério de Lemos, in order to promote the discussion, started the panel by providing a brief introduction to its theme. Open systems, such as the Internet, create conditions where systems can interact and collaborate with one another. Agents, which are autonomous, adaptive and interactive elements with a mental state, can be part of these systems. However, if agent system has to be dependable in the services it provides, additional mechanisms and techniques have to be incorporated in the design of agent systems. In this context, dependability is understood as the ability of an agent or agent system to deliver its service that can justifiably be trusted. However, before starting the theme of the Panel, the moderator provided a brief overview of the previous year's Panel, which was on a related topic "agents and dependability". Three fundamental issues were raised concerning the ability of agents providing dependable services. First, agents have some features that might not be that useful for enabling dependability, for example, autonomy that impair to identify the failure assumptions to be associated with agents, and the mental state of an agent that might restrict the ability for introspection. The latter is particularly fundamental for observing and controlling the state of an agent system in the presence of faults. The second issue raised was related to the role to be played by agents, should it be considered as a basic building block or an additional layer of services? The final issue mentioned was the restrictions that should be imposed on agents for guaranteeing the provision of dependable services, such as, what failure assumptions should be imposed on agent systems for them to reach a distributed consensus? Different from last year's Panel in which the emphasis was on the properties of agents, the theme of this year's Panel was on "agent communities and dependability", in which issues like, collection of agents, communication, and coordination were emphasized. Also the focus of the Panel was on "open systems" rather than architectures, being the latter exemplified by the Open Architecture Agent (OAA), Jade, Zeus, etc. Multi-agent communities in open systems are characterized as heterogeneous since they are created by different people with different intents at different times, using different languages, and autonomous since they have own goals and own thread of control. It should also be considered that in terms of interaction and interoperability, agents join/leave at any time, interact with anyone, and perform any action. In addition, in order to achieve dynamic composition and coordination, agents should rely on contract creation for emulating business relationships. However, there are four essential problematic issues that are associated with multi-agent communities. First, autonomy, how to use, control and manage it? Second, communication, how to ensure interoperability considering that standard protocols might restrict agent autonomy? Third, coordination, how to ensure coherent actions? Finally, the fourth issue is related to knowledge, how to enable automatic and interactive discovery of requirements and instructions? In addition to these well known problems associated with multi-agent communities there is also dependability, which is usually considered as an afterthought in systems design, and which is concerned on how to ensure trust on the services delivered by the system? Taking as basis dependability technologies, which are a collection of methods and techniques by which dependability is attained, the moderator presented some existing attempts in the provision of dependability in multi-agent systems. In the context of rigorous designs, which aims to prevent the occurrence or introduction of faults, there has been some contribution mainly in the area of standards. These standards are mostly related to the semantic Web and Web standards (XML, SOAP, WSDL, UDDI, etc.), and agent standards, such as, the DAML+OIL (DARPA Agent Markup Language Activity), FIPA Agent Communication Language (ACL), Knowledge Interchange Format (KIF), Knowledge Query Manipulation Language (KQML), etc. In addition, there has been some activity in the area of formal approaches to agent-oriented software, methodologies for design and analysis of agent systems, like Gaia and Tropos, and the provision of tool support for modeling and reasoning about security in agent-oriented software engineering. In the context of fault tolerance, which is concerned with the provision of services despite the presence of faults, most of the work has focused on exception handling techniques. The problem with such approach is that application dependent techniques cannot exploit autonomy, and can restrict system coordination. In terms of general solutions for supporting fault tolerance, it is not known of any work that exploits classes of faults. In particular, it is not clear in a multi-agent environment how to reach agreement in the presence of malicious faults, using, for example, group communication algorithms. In terms of verification and validation, which aims to reduce the number and the severity of faults, and system evaluation, which aims to evaluate the presence of faults, their future incidence and consequences, so far there have not been major outcomes. Concerning verification and validation of multi-agents systems, some contributions have been made particularly in testing and model checking. Concerning the latter, the work has been restricted to the behavioral description of the different architectural modules of a single agent. Both contributions are still in their infancy, and they are still dependent on the application domain.

After the brief introduction by the moderator, the questions to be handled by the Panel were presented. The first set of questions was related to the complexity of open systems, and the environment heterogeneity that requires multiple coordination strategies. The main question was how to obtain dependable agent communities for open systems? This question was further partition into three other questions. What are the challenges in terms of dependability technologies during design time (rigorous design, and verification & validation), and run time (fault tolerance)? What are the restrictions that have to be imposed on agents and/or architectures for open systems? How features commonly associated with agents, for example, adaptability, autonomy, learning, mobility etc., can be exploited? The second set of questions was related to fact that agents' autonomy and adaptability should have legal and social implications. The main question was who should be held responsible if an agent fails, causing the services delivered by its community to be catastrophic? This question was further partition into two other questions. What are the safeguards that an agent community should have to deal with agents “misdemeanors”? How a society of agents can tolerate an agent failure, or in other words, what kind of redundancies should be considered in societal terms?

The position taken for the Panel by Gul Agha (University of Illinois, USA) was that dependability is an aggregate property that can only be approximated. Agha started his presentation by saying that dependability is a group property. Dependable individual agents do not imply dependable groups of agents, issues like competition and cooperation between agents have to be considered, and these might affect system dependability. Another issue that is important in the design of multi-agent systems is the representation of aggregate behaviors in the form of parametric models of state. For example, in order to deal with denial of service attacks, the total number of processor cycles should be considered. Another point raised was about controlling resource consumption. This problem can be solved through cybercash, which can be converted into resources. For example, a host provides resources to principal agents, which then distributes the resources between the dependent agents. In the design of dependable complex systems, designers have to pick the worst cases observed and stretch them to some limit. The design of the system should consider these cases, and incorporate safety envelopes. For supporting the process of adaptation for the purpose of obtaining dependable multi-agent systems it is important to learn from other technologies, such as, dynamic program modification, reflection and dynamic adaptation of the environment and evolutionary algorithms.

The position taken for the Panel by Gruia-Catalin Roman (Washington University, USA) was that dependability is the desired outcome of a game played on multiple levels. He begun his talk by presenting what he meant by dependability: understand what is to be feared, compensate for what cannot be controlled, and reason about what will happen. After that, Roman gave his view on open systems operating over ad hoc networks. In terms of system structure, we should consider physical mobility of devices, logical mobility of agents, interactions constrained by logical connectivity, and the availability of resources (data, code, sensors, etc.). In terms of network profile, the essential features of wireless communication should be considered, which might cause, for example, frequent and unpredictable disconnections. In order to deal with these issues Roman proceed to present several design strategies: reduce complexity through coordination, increase productivity through transparent context maintenance, increase stability through stability enhancement, evolve and adapt through code migration. In addition to the above design strategies there are other issues that should be considered: a unified model for reasoning about logical and physical mobility, an abstract treatment of connectivity and access controls, a different way of performing query consistency and guarantees, and a spatiotemporal communication services. He concluded his presentation by stating that dependability is the desired outcome of a game played on multiple levels: the essential traits of the physical environment, the nature of devices and communication, the virtual world offered by the middleware designer, and the predictability achieved through formal analysis.

The position taken for the Panel by Holger Giese (University of Paderborn, Germany) was that safety is different. He started his presentation by analyzing the dependability attributes in the context of multi-agent systems (MAS). MAS can result in high reliability and availability because MAS naturally lead to heterogeneous and redundant processing of tasks. Concerning confidentiality, the design of open agent systems naturally includes the question whether agents can trust each other or not. The strong encapsulation provided by agents builds a natural basis for integrity. Finally, concerning maintainability, the flexibility and adaptability of open agent architectures as well as the strong encapsulation provided by agents simplifies maintenance to a great extend. He argued that although it is a challenging task, multi-agent systems could be made dependable according to the above attributes, however, with safety was different. Giese then enumerated the problems with safe MAS. Open MAS result in more complex or even open systems with more interactions and emergent behaviors. The adaptation and learning aspects lead to non-predictable behaviors. Since safety is a system property, safe agents do not result in safe MAS. Moreover, MAS are only safe with respect to a given environment; hence open MAS and safety do not fit. He then proceeded to explain why safety was different in the context of MAS. MAS are inspired by the intelligent social behavior observed for humans and other natural species that have resulted from (cultural) evolution. Social system "designs" must ensure some degree of reliability, availability; confidentiality, integrity, and maintainability to enable them survive. A social system "design" only survives evolution when it protects enough members from severe and undesirable consequences - one specific design is "safe" for one specific context. The environment will only be protected by a natural social system, while destroying the environment would have severe consequences for the social system itself - safety for the environment occurs only in rare "altruistic" cases. Natural social systems tend to continue operation (being reliable or at least available) under all circumstances while safety-critical systems can behave fail-safe assuming that the operators will restart them. On the other hand, safety is different because safety always includes the members of the society as well as environment, and safety has usually higher demands than only enough surviving society members. In addition, in a safety-critical system catastrophic accidents have to be limited to a probability of at most 10-4 per year (SIL 4), while in natural social systems much lower probability bounds are sufficient to survive. He finished by saying that safety cannot simply be programmed into MAS using the common architectural principles. In his concluding remarks, Giese emphasized three issues: first, that open MAS architectures fit very well to all dependability attributes except safety, second, safety could only be "borrowed" from nature in rare altruistic cases and when the context/design is exactly the same (which usually is not the case!), and thirdly, the requirements for safety-critical systems are much more demanding than the ones in nature! His proposal for intelligent mechatronic agents are: separation of concerns, design for safety, and formal methods can enable us to build safe open MAS systems, if we restrict the relevant concerns in such a manner that we can ensure safety.

P 2: Agent-Oriented Methodologies and Modeling Languages

Chair: Carlos Lucena (PUC-Rio, Brazil)
Panelists: Jaelson Castro, Martin Griss, Tom Holvoet and Arndt von Staa

The moderator, Carlos Lucena, also started the panel by providing a brief introduction to the theme. In recent years agent-oriented methodologies and modeling languages have emerged to provide means for agent-based solution specification. Additionally, agent architectures provide services and infrastructure to agents such as meta-information facility, naming and discovery services and interaction protocols. Hence, software developers can use these predefined services and rely upon the agent architecture to get this support. To start the discussion, the moderator pointed out some of the shortcomings of the existing agent-oriented methodologies and modeling languages. These include: the fact that different existing methodologies, languages and platforms for multi-agent systems propose very distinct and varied sets of abstractions; it is often very difficult to understand the definition of the abstractions and their relationships, and; there is a need to define conceptual framework that define the abstractions and the context the MAS will be inserted.

In fact, the moderator divided the approaches for agent-oriented specification into four categories: (i) agent-oriented conceptual frameworks, (ii) modeling languages that extend object-oriented concepts, such as UML, (iii) modeling languages that are centered in agent abstractions, and (iv) modeling languages that use or extend one of the above. A modeling language is an indispensable element to foster the agent-based software technology. However, this explosion of different methods that rely on different agent concepts make it difficult for developers to start thinking and using the agent abstraction to develop open distributed systems.

After the brief introduction by the moderator, the questions to be handled by the Panel were presented. The first question concerned the agent abstractions. Are they mature enough to develop good agent-oriented methods and modeling languages? This question was further developed into what is it that the methodologies should provide to get the agent technology into the mainstream of software development in industry. Also, there is the concern about agent-oriented systems implementation. How can the gap between specification and code generation, i.e. an agent-based software product, be diminished by these methodologies? The moderator ended his introduction by asking the panelists what were their expectations about the future of research on Agent-oriented Methodologies and Modeling Languages.

The position taken by Jaelson Castro (UFPE, Brazil) was that systems should reflect the underlying organizational/operational environment and that architectures should be designed in terms of the same social and intentional concepts to avoid mismatch. Agent-oriented methodologies may differ in their objectives and underlying premises and also in the way they deal with issues such as openness, uncertainty, security, autonomy, mobility. However, they should cover the different phases of SE and they should have clear guidelines on what decisions are made and when they are made. To do so, he stressed that a requirements-driven development is a good way to specify agent systems, referencing the TROPOS methodology.

Martin Griss (Carnegie Mellon West, USA) started his talk by introducing agents as "flexible components" that are suited for process control and application management, economics (games) and mobile systems. His position was that it is important that modeling languages capture agent roles behaviors and responsibilities. For achieving that, he argued that domain engineering should be used to make specifications at multiple levels of abstraction and with multiple views. He also pointed out that modeling interaction is very important. Well-defined protocols support Model Driven Architecture and protocol descriptions should be reusable artifacts and, most importantly, machine-readable. Griss said that incompleteness and poor documentation are the biggest drawbacks in the existing methods. He also pointed out that agent methodologies should build on mainstream notation (e.g. UML) and process (e.g. RUP, XP).

The position taken by Tom Holvoet (K.U. Leuven, Belgium) was that there are four laws for engineering software for multi-agent systems. The first law is “do not forget you are developing software”. He stated that the research on software engineering for multi-agent systems should be led by mainstream software engineering research and that it should not be considered a completely different field. There are some key cornerstones for developing agent systems such as roles-organizational abstractions, environment specification, architecture definition and actions/perception modeling. The second law is “do not forget you are supposed to develop real software”. There is a big gap between the models and the software artifacts (architectures / frameworks / languages / operating systems /...) to build concrete systems. Any methodology should be supported by (and should be the result of extensive experience with) real-world cases and not just toy applications. The third and fourth laws are concerned with getting the multi-agent system architecture straight. He stated that there is no complete conceptual model for multi-agent system available today and that this is a major topic for future research.

The position taken by Arndt von Staa (PUC-Rio, Brazil) was that graphical modeling languages are insufficient to convey all needed information. Hence, additional information, such as additional representations and structural assertions, is needed. Staa pointed out that representation languages and supporting tools must be designed to adequately help development and maintenance instead of just representing. Development and maintenance is an iterative learning process. Thus, agent-oriented methodologies and modeling languages must be concerned about depicting iterations (or increments) of the knowledge acquired about the system. Evolution is a key aspect of multi-agent systems and successive changes lead to inconsistencies between representations and what is really implemented. He said that all investment in good representations runs a high risk of being wasted if maintenance and evolution are not dealt beforehand.

9. Conclusions

The particular focus of this third meeting was on the role of agents and MAS in open systems. Altogether, the workshop was a very large success due to the quality of the submitted papers, the level of participation of the audience and the profile of the keynote speakers and panelists. SELMAS 2004 achieved its goal to provide a forum for interactive discussions on the research issues of software engineering for large-scale multi-agent systems. The speakers presented items for a research agenda during several of the talks.

SELMAS 2004 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 pull everything together. It is our hope that SELMAS 2004 provided 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.

Like SELMAS, other important, related workshops have been organized to discuss research and practice on multi-agent software engineering (such as AOIS and AOSE workshops [4, 5]). There are a number of ways to learn more about current work and get involved, including:

- Visiting the workshop website for details of ongoing work.
- Reading the position papers from the SELMAS 2002 and SELMAS 2003 LNCS volumes [2, 3] and the SELMAS 2004 proceedings [1] for background information.
- Contacting any of the organizers and authors of the SELMAS papers.

Finally, a high-quality set of workshop and invited papers is going to appear in the third edition of the Software Engineering for Large-Scale Multi-Agent Systems LNCS volume (Springer). The SELMAS 2005 workshop is planned for the next year at ICSE 2005. We look forward to an excellent program also in the next year.

Acknowledgements

We would like to thank all the authors for submitting their work to SELMAS 2004 and we also wish to thank the members of the program committee for their reviews. We would also like to thank the keynote speakers for their special presentations. In addition, we thank the session chairs

and panelists for the fine work in coordinating the sessions and promoting interesting panels, respectively. Finally, we would sincerely like to thank again the SELMAS 2004 participants for their active involvement in the meeting and for the level of their contributions to the debate.

References

[1] Choren, R., A. Garcia, C. Lucena, M. Griss, D. Kung, N. Minsky, and A. Romanovsky (2004): Proceedings of the Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'04). The Institution of Electrical Engineers, Stevenage. ISBN: 0-86341-431-1, 155p.

[2] Garcia, A., C. Lucena, F. Zambonelli, A. Omicini, and J. Castro (2003): Software Engineering for Large-Scale Multi-Agent Systems. Lecture Notes in Computer Science, LNCS 2603. Springer, Berlin. ISBN: 3-540-08772-9, 285p.

[3] Lucena, C., A. Garcia, A. Romanovsky, J. Castro, and P. Alencar (2004): Software Engineering for Multi-Agent Systems II. Lecture Notes in Computer Science, LNCS 2940. Springer, Berlin. ISBN: 3-540-21182-9, 279p.

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

[5] The AOSE International Workshop series, http://www.csc.liv.ac.uk/~mjw/aose/