Editorial

Haralambos (Haris) Mouratidis*

School of Computing and Technology
University of East London
University Way
Beckton, E16 2RD, London
E-mail: H.Mouratidis@uel.ac.uk
*Corresponding author

Marc-Philippe Huget

Polytech'Savoie-LISTIC
University of Savoie
BP 80439
74944 Annecy le Vieux Cedex, France
E-mail: Marc-Philippe.Huget@univ-savoie.fr

Biographical notes: Dr. Haralambos (Haris) Mouratidis holds a BEng in Electronics with Computing Science from the University of Wales, Swansea, UK, and MSc and PhD degrees from the University of Sheffield, UK. He is currently a Senior Lecturer in Computer Science at the School of Computing and Technology at the University of East London. His research interests lie in the areas of agent-oriented software engineering, security engineering and information systems engineering and he has published more than 60-refereed papers related to these research areas. He has been actively involved in the organisation of research events such as Agent Oriented Information Systems (AOIS) and Safety and Security in MultiAgent Systems (SASEMAS) and he has served on a number of programme committees, such as Autonomous Agents and MultiAgent Systems (AAMAS), Conference on Advanced Information Systems Engineering (CaiSE), Intelligent Agent Technology (IAT) and Agent-Oriented Software Engineering (AOSE).

Dr. Marc-Philippe Huget holds a PhD in Computer Science from the University of Paris Dauphine, and is currently the Associate Professor at Polytech'Savoie from the University of Savoie, France. He has been working actively on multi-agent systems since 1997, mainly on agent-oriented software engineering. His main interests are on agent methodologies, notations and programming languages. He is known for his work on agent Unified Modelling Language (UML) modelling notation and the two specifications he co-authors as part of the Foundation for Intelligent Physical Agents (FIPA) standardisation effort. His current research is about model-driven engineering and domain-specific languages for multi-agent system development.

Over the past decade, software agents and multiagent systems have grown into one of the most active areas of research and development in computing. At the same time, they affect other computing areas such as the grid, the SemanticWeb and WebServices. Plenty of reasons have been given in the computing literature why agents have received such interest; certainly one of the most important reasons is that the concept of an agent as an autonomous system, capable of interacting with other agents in order to satisfy its design objectives, provides a natural way of developing software. An agent-oriented approach views the system as a society, similar to a human society, consisting of entities that possess characteristics similar to humans such as mobility, intelligence and the capability of communicating.

As a result, there has been a growth of interest in the potential of agent technology in the context of software engineering. Significant research has been introduced under the umbrella term of Agent-Oriented Software Engineering (AOSE), which contains both theoretical and practical work for agent systems in areas such as software development methods, software methodologies, modelling languages and notations, computer-aided software engineering tools, and implementation and execution platforms.

This special issue is focusing in the area of modelling languages for multiagent systems; an important, active and fast moving line of research associated with agent oriented software engineering. In the context of agent oriented software engineering, modelling languages provide a collection of elements that assist the modelling and description of multiagent systems, in terms of different artefacts, across different stages of the development cycle such as analysis, design, development, test and deployment.

There are several modelling languages in the context of multiagent systems. Although the history of the research efforts to construct such languages is not long, we can differentiate between important periods. During the early and middle 1990s, when the first multiagent systems were developed, researchers focus less on the definition of engineering approaches (such as methodologies, modelling languages and so on), and more on the usage of formal theory behind the development of multiagent systems. From late 1990s the importance of developing structured approaches towards the engineering of multiagent systems is realised and research highlights the importance to propose modelling languages specific to multiagent concepts by building on existing industrial languages. A number of 'first generation' modelling languages for agent systems appear mainly based on existing object-oriented languages. These efforts are restricted by the limited support for agent specific concepts, lack of tools, documentation and industrial cases studies to support the applicability of the languages. More recently, and since a better understanding of agent oriented software engineering is slowly gained, researches have recognised that the realisation of the long term aim to develop mature modelling languages for multiagent systems implies the development of modelling languages that are capable of modelling every aspect of multiagent systems; are supported by appropriate documentation and software tools; and are applicable to a wide range of industrial complex applications.

The aim of this special issue is to provide a forum for representing the most recent and innovative lines of research related to modelling languages for agent systems and capture its essential elements. Twelve papers were submitted, as a result of the CFP, discussing research advances in a range of topics relevant to the special issue, such as new modelling language proposals, formalisation of modelling languages, pitfalls and

lessons learned in the development of modelling languages, comparative studies and industrial cases. After two rounds of reviews three papers were accepted to be published in this special issue.

In 'MAS-ML: a multiagent system modelling language' by da Silva *et al.*, the authors argue that UML cannot be used for agent-oriented modelling directly or with simple, narrow changes but a rational set of modifications to the UML metamodel is needed so that it can become more appropriate for multiagent systems development. The authors present a modelling language, called MAS-ML, based on extensions to the UML 2.1 metamodel. MAS-ML differs from the existing UML-based approaches because it explicitly introduces new concepts to the UML metamodel to deal with agent-oriented entities, behaviour abstraction and decomposition, social and mental aspects, and communicative interactions. In doing so, it incorporates agent-related abstractions such as roles, organisations, plans and protocols and it defines a set of new modelling elements and a set of structural and dynamic diagrams. In 'Automated testing sequences generation from AUML diagrams: a formal verification of agent interaction protocols' by Mokhati *et al.*, a novel technique is presented to support the automatic generation of testing sequences from Agent UML (AUML) – Agent Interaction Protocol (AIP) – diagrams. The verification of the interactions between agents relies on the generation of a specification of the agents of the multiagent system based on the formal specification language Maude. Such specification is automatically generated from the appropriate AUML diagrams. The presented approach is based on two main steps, the generation of reduced testing sequences and the generation of complete testing sequences. The authors have also developed a set of tools to support the different stages of their approach. In 'Formally specifying and verifying mobile agents: model checking mobility – MobiOZ approach', by Taguchi *et al.*, an extension of the Object-Z formal notation, called MobiOZ, is presented. MobiOZ extends Object-Z with mobile and communication primitives that assist the specification and verification of mobile agent applications. Moreover, the authors present its semantic foundation along with a verification based method that details how specifications in MobiOZ can be simulated and verified with the aid of a model checking tool. The applicability of the proposed formal notation and verification method is illustrated with the aid of a set of examples.

We would like to thank all the authors for their contribution, the reviewers for their professionalism and the quality of their reviews, and also the Editors-in-Chief of the *International Journal of Agent Oriented Software Engineering* for the opportunity to publish this issue and for their support.