Auton Agent Multi-Agent Syst (2011) 23:155–157
DOI 10.1007/s10458-010-9162-1

Preface

Rafael H. Bordini · Mehdi Dastani · Jürgen Dix · Amal El Fallah Seghrouchni

Published online: 6 January 2011

This special issue of the *Journal for Autonomous Agents and Multiagent Systems* contains selected papers from the Dagstuhl Seminar 08361 on "Programming Multi-Agent Systems", which was held in Dagstuhl, Germany, from August 31st to October 5th, 2008.

Intelligent, autonomous agents and particularly Multi-Agent Systems (MAS) play an important role in current trends in software development. Indeed, they constitute a new and interesting paradigm to implement complex systems: they propose relevant abstractions for software engineering in challenging areas of application. Several application domains, some at industrial level, take benefit from MAS technology.

For almost two decades, the MAS community has developed and offers a large and rich set of concepts, architectures, interaction techniques, and general approaches for the analysis and the specification of MAS. It still remains a challenge to consolidate the existing practical tools and techniques in order to provide practitioners with mature languages and platforms that can help them design, implement and deploy efficiently a new generation of complex software built as MAS. The organisers of Dagstuhl Seminar 08361 as well as the participants share the conviction that the success of agent-oriented systems depends on expressive programming languages and well-developed platforms being available so that the concepts and techniques of multi-agent systems can be easily and directly implemented.

The aim of this seminar was to bring together researchers from both academia and industry for bridging this gap and identifying interesting lines of research within multi-agent systems. The topics ranged from *theoretical foundations*, *programming languages*, to *development tools and applications*.

The four papers selected for this special issue (selected from 8 submissions that were invited from among the best contributions of the Dagstuhl seminar), listed below, were carefully reviewed by expert referees. They constitute improved versions of papers presented at the seminar.

- In *Environment Programming in MAS—An Artefact-based Perspective* by Alessandro Ricci, Michele Piunti, and Mirko Viroli, the authors present a programming framework

R. H. Bordini · M. Dastani · J. Dix (✉) · A. E. F. Seghrouchni
Clausthal University of Technology, Clausthal-Zellerfeld, Germany
e-mail: dix@tu-clausthal.de

for the development of the environment component of multi-agent systems. The shared environment in a multi-agent system is considered as a workspace consisting of a dynamic set of entities called artefacts. Each artefact represents a tool or a resource that is used and shared by individual agents at run time. The artefacts can be used by the agents in order to share information or coordinate their activities. The paper explains and discusses the underlying concepts related to artefacts and describes a technology called CArtAgO which can be used to develop shared environments for multi-agent system.

- In *Engineering and Verifying Agent-Oriented Requirements augmented by Business Constraints with B-Tropos* by Marco Montali, Paolo Torroni, Paola Mello, Nicola Zannone, and Volha Bryl, the authors present B-Tropos, which can be seen as a modeling framework to support agent-oriented systems engineering, from high-level requirements elicitation down to execution-level tasks. B-Tropos extends the original Tropos methodology by means of declarative business constraints, inspired by the ConDec graphical language. The functioning of B-Tropos is illustrated using an example inspired by a real-world industrial scenario. The paper also shows how B-Tropos models can be automatically formalized in computational logic, and it discusses some formal properties of the resulting framework and its verification capabilities.
- In *Declarative Programming for Agent Applications*, by John Lloyd and Kee Siong Ng, the authors provide a formal account of the execution model of the Bach programming language that combines both the logical and functional paradigms. The language is based on higher-order logic and modal-logic programming as well as probabilistic computation, and has some theorem proving and equational reasoning capabilities. It has been implemented on top of Haskell. The multi-modal and probabilistic dimensions are particularly targeted at agent applications, as illustrated by examples in the paper.
- In *Specifying Recursive Agents with GDTs*, by Bruno Mermet and Gaële Simon discuss the GTD formalism which allows the specification of goal decomposition by means of various operators. LTL is used to define the semantics of that formalism and there is also a graphical notation to facilitate design. Properties such as invariants at the agent and system levels can be proved for a specification, from which code can be automatically generated. The paper introduces design patterns and proof schemas for agents that are defined by composing various GDTs.

We are grateful for having had the opportunity to organise this fruitful seminar at Dagstuhl. A similar seminar helped us, six years ago, to start a successful international cooperation in the domain of Multi-Agent Programming (in particular the ProMAS workshop series, see bibliography). We are equally grateful to having been offered by the editors-in-chief of the *Journal of Autonomous Agents and Multiagent Systems* a special issue for the best papers based on talks given by the Dagstuhl seminar participants.

Finally, we would like to thank all authors for their contribution to this special issue, and we are much grateful to all the reviewers for their thorough work that allowed us to select four papers that represent the breadth of the ideas discussed at the Dagstuhl Seminar.

References

1. Dastani, M., Dix, J., & Seghrouchni, A. E. F. (Eds). (2004). Programming multi agent systems. *ProMAS '03, post-WS proceedings. Lecture notes in artificial intelligence* (Vol. 3067). NY: Springer.
2. Bordini, R., Dastani, M., & Dix, J., Seghrouchni, A. E. F. (Eds). (2005). Programming multi agent systems. *ProMAS '04, post-WS proceedings. Lecture notes in artificial intelligence* (Vol. 3346). NY: Springer.

3. Bordini, R., Dastani, M., Dix, J., & Seghrouchni, A. E. F. (Eds). (2005). Programming multi agent systems: Languages, platforms and applications, multiagent systems. *Artificial societies, and simulated organizations* (Vol. 15). NY: Springer.
4. Bordini, R., Dastani, M., Dix, J., & Seghrouchni, A. E. F. (Eds). (2006). Programming multi agent systems. *ProMAS '05, post-WS proceedings. Lecture notes in artificial intelligence* (Vol. 3862). NY: Springer.
5. Bordini, R., Dastani, M., Dix, J., & Seghrouchni, A. E. F. (Eds). (2007). Programming multi agent systems. *ProMAS '06, post-WS proceedings. Lecture notes in artificial intelligence* (Vol. 4411). NY: Springer.
6. Bordini, R., Dastani, M., Dix, J., & Seghrouchni, A. E. F. (2007). Programming multi-agent systems. *Special Issue of International Journal of Agent-Oriented Software Engineering, 1*(3).
7. Bordini, R., Dastani, M., Dix, J., & Seghrouchni, A. E. F. (Eds.). (2009). *Multi-agent tools: Languages, platforms and applications*. NY: Springer.