Contents lists available at SciVerse ScienceDirect

Science of Computer Programming

journal homepage: www.elsevier.com/locate/scico

Editorial

Preface to the special section on agent-oriented design methods and programming techniques for distributed computing in dynamic and complex environments

Agent-based approaches have become quite popular as design methods and programming techniques for, broadly understood, distributed computing. In particular, they turn out to be effective in the development of systems for dynamic and complex environments, where they can provide the needed flexibility and robustness. In this context, we can also observe that the field of agent technology is maturing. One of the key factors that influence this process is the gathered body of knowledge that allows ever more complete reflection on the nature of design and implementation of agent systems.

These processes can be observed at all levels of the "software stack" represented through the specific topics covered in this special section. First, we can see the emergence of new *formalisms* for agent-based systems, for instance the AML or the API Calculus. Furthermore, existing methodologies for agent system design and implementation are starting to be supported by *tools* that not only allow top level (proto)formalization, but also guide the design process from the initial conceptualization towards the implementation (e.g. by generating foundations of the initial code). Separately, existing tools/platforms/environments have evolved through a number of releases, eliminating early stage problems and adding a number of new, important, features. Resulting products are becoming truly robust and flexible. Finally, open source products (e.g. JADE) are surrounded by user communities, which often generate powerful add-on components, further increasing the value of the existing solutions. Through all these processes, combined with an increasing number of demonstrator systems, we gain in-depth understanding of most important issues in the general area of Agent-Oriented Software Engineering (AOSE). These issues will have to be addressed by future *methodologies, formalisms, languages*, and *tools* developed to support the design and implementation of agent systems.

In this context, the workshop Agent Based Computing – from Model to Implementation (ABC:MI) held at Wisła, Poland, on 18–20 October 2010, was a very interesting opportunity to involve researchers in the agent area in an in-depth discussion of their research work. The peculiarity of this workshop was its call for papers that explicitly connect the design aspects to the implementation ones. From this workshop, a need arose for a special section of a high quality journal, highlighting the current situation and reflecting the state of the art.

Starting from these considerations, this special section of the journal Science of Computer Programming aims at proposing an up-to-date picture of the situation in AOSE in connection with the previously mentioned topics, by means of presenting high quality papers that show the state of the art in research work related to a pathway from the model of the problem domain to the actual agent-based solution. We invited the submission of the best papers from the ABC:MI workshop, and, at the same time, we made an open call in order to gather additional high quality papers. The call attracted 18 papers. After a thorough refereeing process, seven of them were selected for publication. In the following, readers can find a brief summary of each paper.

In "Towards a next-generation AOSE methodology", Michael Winnikoff and Hoa Khanh Dam consider the existence in the literature of a very large number of methodologies for the developing of agent systems. Their sheer number creates a challenge to practitioners, who need to select a methodology to adopt in their project(s). This situation seems to be analogous to that of object-oriented methodologies and notations before the successful introduction of the UML, and the authors argue that the time is ripe to begin the development of a next generation Agent-Oriented Software Engineering (AOSE) methodology, with the ultimate goal of the creation of a unified AOSE methodology. This paper outlines processes and models for such a next generation AOSE methodology. The authors' proposal is based on a comparative analysis of seven prominent AOSE methodologies, with identified strengths, weaknesses, commonalities and differences.

"A model-driven CASE tool for developing and verifying regulated open MAS", by Emilia Garcia, Adriana Giret, and Vicente Botti, concerns the development of agent-based complex systems. In particular, the authors propose a CASE tool for supporting the agent developers in their jobs. Such a tool is based on model-driven technologies in order to integrate

doi:10.1016/j.scico.2012.01.001

the design of the agent-based complex systems with the verification of their models; eventually, the tool produces part of the code of the system to be implemented.

"Simulation in Agent-Oriented Software Engineering: The SODA case study", by Ambra Molesini, Matteo Casadei, Andrea Omicini, and Mirko Viroli, focuses on Agent-Oriented Software Engineering (AOSE), proposing a simulation-based approach to Multi-Agent System (MAS) engineering. MAS are complex systems that require a careful development to be effective. In this paper, the authors show how a simulation-based approach to MAS engineering can be integrated within existing AOSE methodologies, in order to enable engineers, by means of simulation, to foresee at design time the properties of the implemented systems. The SODA methodology is considered as a case study.

"A scalable multiagent system architecture for interactive applications", by Guillermo Vigueras, Juan Manuel Orduña Huertas, Miguel Lozano, and Yvon Jègou, addresses an important field inside the agent area, the one of the simulation of complex environments. Agents are particularly useful for simulating crowded situations, where each agent can represent one person, with common and different features among agents. The authors present a distributed multi-agent architecture that can manage large crowds of autonomous agents and can provide multiple views of the simulated virtual world. Such an architecture can be exploited to implement a range of simulation applications. One of the interesting aspects of this paper is that the proposed architecture was designed to enable a high degree of scalability in implemented applications.

"An application-level technique based on recursive hierarchical state machines for agent execution state capture", by Giancarlo Fortino and Francesco Rango, focuses on the capturing of the agent state. This turns out to be useful for different purposes in dynamic applications based on agents. The proposed approach is based on the Distilled StateCharts Star (DSC*) formalism that offers an agent-oriented type of recursive hierarchical state machines. This technique can be applied to agents implemented in different programming languages and for different platforms.

"Multi-agent oriented programming with JaCaMo" by Olivier Boissier, Rafael H Bordini, Jomi F Hubner, Alessandro Ricci, and Andrea Santi, aims at connecting different programming paradigms related to the development of dynamic and complex systems: agent-oriented programming, organization-oriented programming, and environment-oriented programming. The idea is made concrete in the proposal of JaCaMo, a platform that integrates three existing platforms in order to join their functionalities. So, the design of the different aspects of a complex agent-based system can rely on the implementation by a unified platform.

"A multi-agent approach to distributed ant colony optimization", by Costin Badica and Sorin Ilie, proposes a configurable distributed architecture for addressing the ant colony optimization problem. In such a problem, the environment can be dynamic and complex, and is managed by a large amount of autonomous agents. The proposed approach relies on asynchronous messages exchanged by agents, which provide effectiveness and scalability to the applications that can be implemented exploiting this architecture.

We hope that interested readers can find in these papers useful hints for their research. Last but not least, we would like to thank the editor of Science and Computer Programming, not only for the chance of editing a special section, but also for all the highly valuable support he gave us during the entire process; we are also very grateful to the reviewers, who spent valuable time on ensuring the quality of the published papers; finally, we would like to thanks all the papers' authors, both accepted and not, for having considered this special section as a means for disseminating their research work.

Special section guest editors
Giacomo Cabri
University of Modena and Reggio Emilia, Italy

Maria Ganzha*
System Research Institute, Polish Academy of Sciences, Warsaw, Poland
University of Gdańsk, Poland
E-mail address: Maria.Ganzha@ibspan.waw.pl.

22 December 2011
Available online 6 February 2012

* Corresponding editor.