Editorial

Danny Weyns

DistriNet Labs
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3001 Leuven, Belgium
E-mail: danny.weyns@cs.kuleuven.be

Biographical notes: Danny Weyns is a post-doctoral researcher at the Katholieke Universiteit of Leuven, Belgium. He obtained his PhD in 2006 for research on the connection between multiagent systems and software architecture. Weyns's research interests include the role of the environment in multiagent systems, architectural description languages for decentralised systems and aspect-oriented software development.

Since the mid-1990s the idea that multiagent systems are a *radically new* way of engineering software has dominated research in agent-oriented software engineering. Wooldridge *et al*. (2000) state:

> "There is a fundamental mismatch between the concepts used by object-oriented developers and other mainstream software engineering paradigms, and the agent-oriented view. [...] Existing software development techniques are unsuitable to realise the potential of agents as a software engineering paradigm."

Zambonelli and Omicini (2003) state, "Agent-based computing can be considered as a new general-purpose paradigm for software development, which tends to radically influence the way a software system is conceived and developed." This vision has led to the development of numerous multiagent system methodologies. However, the results that have been obtained in the application of these methodologies to real-world problems are disappointing.

Another perspective on engineering multiagent systems starts from the viewpoint that multiagent system engineering fits well within mainstream software engineering. One particular argument that researchers of this vision have put forward is the observation that there is a strong connection between multiagent systems and software architecture. Mainstream software engineering has generally recognised software architecture as the primary vehicle to manage complexity and to achieve the system's required qualities. Software architecture consists of the structures of the system, which comprise software elements, the externally visible properties of those elements and the relationships among them (Bass *et al*., 2003). Software elements provide the functionality of the system, while the required system qualities are primarily achieved through the structures of the software architecture. A multiagent system is structured as a set of autonomous software components (agents) that are situated in a shared environment. Agents can flexibly

Copyright © 2008 Inderscience Enterprises Ltd.

achieve their design objectives by acting in the environment and interacting with one another. Multiagent systems are typically ascribed quality properties such as adaptability, robustness and scalability.

The connection between multiagent systems and software architecture was the subject of a special track at Net.ObjectDays 2006 (Weyns and Holvoet, 2006). This special issue contains a selection of four thoroughly revised papers from this special track. 'View composition in multiagent architectures' by Boucké and Holvoet argues for explicit support in relating and composing multiple views in architectural descriptions. The authors propose three types of relations and demonstrate their use in an industrial multiagent system. In 'On the modularity assessment of aspect-oriented multiagent architectures: a quantitative study', Sant'Anna *et al*. analyse two medium-sized multiagent systems. A quantitative comparison demonstrates the advantages of improved modularity of aspect-oriented multiagent system architectures over conventional multiagent system architectures. In 'Engineering manufacturing control systems using PROSA and delegate MAS', Verstraete *et al.* present a software product line architecture for agent-based manufacturing control. The authors describe the assets of this software product line architecture and show how these assets can be combined to instantiate concrete manufacturing control applications. Finally, 'Architectural design of a situated multiagent system for controlling automatic guided vehicles' by Weyns and Holvoet demonstrates an end-to-end approach to applying software architecture principles in the design of an industrial AGV transportation system that is structured as a situated multiagent system.

The papers of this special issue show how the architectural design of multiagent systems fits well within a mainstream software engineering perspective. They illustrate how the cross-fertilisation of multiagent system engineering and mainstream software architecture paves the way for industrial acceptance of multiagent systems.

Acknowledgements

I am grateful to IJAOSE and its Editors-in-Chief Brian Henderson-Sellers and Paolo Giorgini for offering the opportunity to publish this special issue. Many thanks also to the anonymous reviewers for their valuable comments.

References

Bass, L., Clements, P. and Kazman, R. (2003) *Software Architecture in Practice*, Addison Wesley Publishing Comp., ISBN 0321154959.

Weyns, D. and Holvoet, T. (2006) 'Multi-Agent Systems and Software Architecture (MASSA)', *Proceedings of the Special Track at Net.ObjectDays 2006*, K.U. Leuven, ISBN 9056827375.

Wooldridge, M., Jennings, N. and Kinny, D. (2000) 'The Gaia methodology for agent-oriented analysis and design', *Autonomous Agents and Multi-Agent Systems*, Vol. 3, No. 3, pp.285–312.

Zambonelli, F. and Omicini, A. (2003) 'Challenges and research directions in agent-oriented software engineering', *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 9, No. 3, pp.253–283.