

# Current Issues in Multi-Agent Systems Development

Rafael H. Bordini<sup>1</sup>, Mehdi Dastani<sup>2</sup>, and Michael Winikoff<sup>3</sup>

<sup>1</sup> Department of Computer Science  
University of Durham  
`r.bordini@durham.ac.uk`

<sup>2</sup> Institute of Information and Computing Sciences  
Utrecht University  
`mehdi@cs.uu.nl`

<sup>3</sup> School of Computer Science and Information Technology  
RMIT University  
`winikoff@cs.rmit.edu.au`

Multi-agent Systems are seen as being an appropriate technology to develop complex distributed software systems. The development of such systems requires a methodology that can guide developers to build and maintain such complex software systems. From the software engineering perspective, the systematic development of software should follow several phases. The number and purpose of the phases depend on the problem domain and the methodology. In general, it is easy to find requirement elicitation, system analysis, architectural design, implementation, and validation/verification phases in most software development methodologies.

Various agent-oriented methodologies have been proposed, such as GAIA (Zambonelli *et al.* 2003), Prometheus (Padgham and Winikoff 2004), Tropos (Bresciani *et al.* 2003), MaSE (DeLoach *et al.* 2001), SODA (Omicini 2000), MAS-CommonKADS (Iglesias *et al.* 1998), and INGENIAS (Gomez-sanz *et al.* 2003). Each of these methodologies focuses on the role of agent abstractions on certain phases of the multi-agent system development. However, the main focus of most of these agent-oriented methodologies has been on the analysis and design phases, paying less attention to the implementation and testing of multi-agent systems. Another active area of research is that concerned particularly with programming languages for multi-agent systems and platforms and tools for multi-agent systems development (Bordini *et al.* 2005).

In our view, the development of multi-agent systems involves dealing with various concerns such as individual agents, their organisations and coordination mechanisms, and shared environments. According to this view, a mature development methodology for multi-agent systems should provide guidelines, concepts, methods, and tools to facilitate the development of all these concerns. In this paper, we discuss the progress so far and future issues related to these concerns in multi-agent systems development, from three main perspectives: design, implementation (programming languages in particular), and verification. For each of these perspectives, we provide a summary of existing approaches and the open issues that require urgent solutions.

## References

- Bordini, R. H., Dastani, M., Dix, J. and El Fallah Seghrouchni, A. editors. 2005. *Multi-Agent Programming: Languages, Platforms and Applications*. New York: Springer.
- Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J. and Perini, A. 2003. TROPOS: An agent-oriented software development methodology. *Journal of Autonomous Agents and Multi-Agent Systems*, **8**(3), 203–236.
- DeLoach, S. A., Wood, M. F., and Sparkman, C. H. 2001. Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering* **11**(3), 231–258.
- Gomez-Sanz, J., and Pavon, J. 2003. Agent oriented software engineering with INGENIAS. In *Lecture Notes in Computer Science, 2691*. Berlin: Springer, pp. 394–403.
- Iglesias, C. A., Garijo, M., Centeno-Gonzalez, J., and Velasco, J. R. 1998. Analysis and Design of Multiagent Systems Using MAS-Common KADS. In *Lecture Notes in Computer Science, 1365*. London: Springer-Verlag, pp. 313–327.
- Omicini, A. 2000. SODA: Societies and infrastructures in the analysis and design of agent-based systems. In *Lecture Notes in Computer Science, 1957*. Berlin: Springer, pp. 185–193.
- Padgham, L. and Winikoff, M. 2004. *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley and Sons, Chichester, UK. ISBN 0-470-86120-7.
- Zambonelli, F., Jennings, N. R. and Wooldridge, M. 2003. Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, **12**(3), 317–370.