

Issues in Multiagent System Development¹

Mehdi Dastani Joris Hulstijn Frank Dignum
John-Jules Ch. Meyer
ICS, Utrecht University
{mehdi,jorish,dignum,jj}@cs.uu.nl

In this paper we discuss development methodologies for multiagent systems. Besides guidelines for the analysis and design phase, a methodology should also provide guidelines for the implementation phase, and explain how the design concepts can be mapped onto instructions of an available programming language. For applications that are best understood in terms of norms and social structures, that implies that a methodology should indicate in which ways norms can be designed and implemented. For example, a general norm can be hardwired into the environment, but can also be translated into norm-abiding goals of the individual agents. This choice has important consequences for the internal design of an agent. Therefore we believe that a methodology should not only consider inter-agent, but also intra-agent aspects.

Issues We selected four dedicated agent-oriented software methodologies and made a review. The methodologies are Gaia [8], AAIL [6], SODA [7], and Tropos [1]. We identified a number of issues that are problematic for these methodologies.

- There is no agreement on how to identify and characterize *roles* in the analysis phase and *agent types* in the design phase.
- The concepts used in the methodologies, like responsibility, permission, goals and tasks do not have a *formal semantics* or *explicit formal properties*.
- There is a *gap* between the *design models* of the methodologies and the existing *implementation languages*. To bridge the gap, a methodology should either introduce design models that can be directly implemented, or use a dedicated agent-oriented programming language.
- The methodologies that do include an implementation phase, such as Tropos, do not explain how to implement reasoning about *beliefs*, reasoning about *goals* and *plans*, reasoning about *planning goals*, or reasoning about *communication*.
- It is widely recognized that an agent may enact several roles. None of the methodologies address reasoning about *playing different roles*.
- The methodologies, with the exception of [8], ignore *organizational norms* and do not explain how to specify and design them.
- *Open systems* are not really supported. The methodologies implicitly suppose that agents are purposely designed to enact roles in a system. But as soon as agents

¹This abstract summarizes [3].

from the outside may enter, the analysis, design and implementation need to treat agents as given entities.

- In the analysis, methodologies do not consider the *environment* in which a system will be embedded. The structure of the organization in which a system will be embedded, has a large influence on the type of organizational structure of the system.

Towards a Solution To overcome some of these problems, we propose an alternative multiagent methodology. The analysis phase is based on OperA [5], which captures, among other aspects, the notion of norms and organisation structure. The design phase is based on the design models of the methodologies discussed above, extended with design models of individual agent types (intra-agent design). The implementation phase is based on the 3APL programming language [2, 4], which provides programming constructs to capture the intra-agent issues, social and cognitive concepts, and the multiagent environment.

References

- [1] J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven information systems engineering: the TROPOS project. *Information Systems*, 27:365–389, 2002.
- [2] Mehdi Dastani, Frank de Boer, Frank Dignum, and John-Jules Meyer. Programming agent deliberation. In *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'03)*. 2003.
- [3] Mehdi Dastani, Joris Hulstijn, Frank Dignum, and John-Jules Meyer. Issues in multiagent system development. In *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*. 2004.
- [4] Mehdi Dastani, Birna van Riemsdijk, Frank Dignum, and John-Jules Meyer. A programming language for cognitive agents: Goal directed 3apl. In *First Workshop on Programming Multiagent Systems (ProMAS'03)*. 2003.
- [5] Virginia Dignum. *A Model for Organizational Interaction, based onAgents, founded in Logic*. PhD thesis, University of Utrecht, 2003.
- [6] D. Kinny, M. Georgeff, and A. Rao. A methodology and modelling technique for systems of bdi agents. In *Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-96)*, LNCS 1038. Springer Verlag, 1996.
- [7] Andrea Omicini. SODA: Societies and infrastructures in the analysis and design of agent-based systems. In *AOSE*, pages 185–193, 2000.
- [8] F. Zambonelli, N. Jennings, and M. Wooldridge. Organizational abstractions in the analysis and design of multi-agent systems. In *First International Workshop on Agent-Oriented Software Engineering at ICSE*. 2000.