Moving Multiagent Systems from Research to Practice

By Scott A. DeLoach, Kansas State University

The state-of-the-art in multiagent research and engineering is insufficiently reflected in state-of-the-practice in complex distributed systems for the basic reason that we have yet to demonstrate, or at least publicize, the significant benefits of using true agent-oriented approaches to solve complex problems. I believe that many practitioners do not see the multiagent approach as being technically superior; for every multiagent system that achieves success, it is possible to envision a non-agent approach that is equally suited for the task. After all, almost all agent systems are programmed in the same programming languages as non-agent systems. What we have failed to demonstrate is that the agent approach can yield technically competitive (or better) solutions with a real benefit, most likely in terms of reduced costs, greater reliability, greater flexibility, or a greater chance of repeatable success. Agent-oriented software engineering lies directly at the heart of this problem. What we need to show is that we can build reliable complex, distributed systems using agent-oriented approaches that are repeatable and sound.

However, there are currently several obstacles that hamper progress towards being able to use multiagent systems and agent-oriented software engineering in mainstream applications. These include

1. the lack of a common understanding of key multiagent concepts,
2. the lack of a set of common notations and models, and
3. the lack of flexible, industrial strength methods and techniques for developing multiagent systems.

The lack of an agreement on the key multiagent concepts and their definitions is the first obstacle to be breached in the battle toward making multiagent systems a mainstream paradigm. For instance, the vast majority of computer science students and practicing professionals would be easily able to define and generally agree upon the basic definitions of the object-oriented notions of objects, classes, generalization, specialization, and aggregation. Yet, at the same time, most experienced multiagent researchers would have a difficult time trying to reach agreement on the commonly used notions of agents, roles, conversations, plans, organizations, or capabilities. The closest thing we have to agreement is on the definition of an *intelligent agent* as a computational system that senses and acts autonomously in a dynamic environment in order to realize a set of goals [6]. Although many researchers and practitioners use the names to represent similar concepts, the real problem lies in the relationships between the concepts.

A second major obstacle I see is the lack of a common notation and models for multiagent concepts. Of course, given that we have not decided on the definition of the concepts and their relationships themselves, finding a common representation may seem like an insignificant problem. However, a lack of a common notation makes it hard for practitioners to investigate different methods and techniques since they have to relearn notation for each different approach. Also, a common notation makes the similarities between approaches and models much easier to spot. In recent work with Lin Padgham and Michael Winikoff, we found that after putting our respective set of models (O-MaSE [5] and Prometheus [4]) into a common notation, the similarities between the two methodologies and the concepts we used was much more readily apparent.

The third obstacle is the lack of strong industry acceptance for any current agent-oriented methodologies. Reasons for this lack of acceptance include the variety of concepts and approaches upon which these methodologies are based along with a lack of tools to support them. However, I believe that one of the major reasons for this lack of acceptance is that the current set of methodologies tends to be inflexible and hard to extend for a variety of applications. One solution is to allow users to customize methodologies to the different types of applications being developed. There have been some suggestions for increasing industrial acceptance. For instance, Odell et al. suggest presenting new techniques as an incremental extension of known and trusted methods [3], Bernon et al. suggest the integration of existing agent-oriented methodologies into one highly defined methodology [1], and Henderson-Sellers suggests the use of method engineering [2].

Based on these observations, I think that agent-oriented software engineering researchers and multiagent systems researchers must address the obstacles address above. We must work on defining a core set of concepts that are well understood and accepted amongst all multiagent practitioners. Essentially, this boils down to defining a core metamodel for multiagent systems. While not all concepts and relationships need be represented, the core concepts and their relationships must be defined. There has been work toward defining a common [1], unfortunately, the proposed metamodels tend to be overly complex and of limited practical use. Once this first step is in place, the next two steps, creating a common notation, and creating industrial strength methods and techniques can be pursued. Finally, having a industrial strength methods and techniques in place will enable the ultimate goal of demonstrating the usefulness of multiagent approaches in the development of sound and repeatable complex, distributed systems.

[1] Bernon C., Cossentino M., Gleizes M., Turci P., and Zambonelli F.: A study of some multi-agent meta-models. In: Odell, J., Giorgini, P., and Müller, J. (eds.) Agent Oriented Software Engineering V. LNCS 3382. Springer-Verlag, Berlin Heidelberg New York (2004) 62–77.

[2] Henderson-Sellers, B., and Giorgini P. (eds.): Agent-Oriented Methodologies, Idea Group Inc., 2005.

[3] Odell J., Parunak V. D., and Bauer B.: Representing Agent Interactions Protocols in UML. In: Ciancarini, P., and Wooldridge, M. (eds.): Agent Oriented Software Engineering. LNCS. 1957. Springer-Verlag, Berlin Heidelberg New York (2001) 121–140.

[4] Padgham, L. and Winikoff, M. *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley and Sons, 2004. ISBN 0-470-86120-7.

[5] Garcia-Ojeda, J.C., DeLoach, S.A., Robby, Oyenan, W.H., and Valenzuela, J. O-MaSE: A Customizable Approach to Developing Multiagent Development Processes. In Luck, M., and Padgham, L. (eds.), *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent Oriented Software Engineering* (AOSE 2007), LNCS 4951, 1-15, Springer-Verlag: Berlin.

[6] Russell, Stuart J. & Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2