

Concepts, Method Engineering and Future Standardization for AOSE.

Brian Henderson-Sellers, Professor of Information Systems,
Faculty of Information Technology, University of Technology, Sydney, Australia

The challenges for agent-oriented software engineering include, in my opinion:

- The lack of a good example. Most exemplars I see whilst reviewing articles use system designs that I could easily accomplish just as well using an object-oriented development approach. That means that potential users of AOSE are not persuaded to spend their precious time in getting to grips with the new ideas of AOSE.
- The persisting gap between agents in the classical or AI-influenced sense and true AOSE. At agent-oriented conferences like AAMAS, the majority of attendees more readily fit into the first category. To encourage more AOSE, there are moves afoot to create a truly software engineering flavoured agent conference as well as our initiative to create an AOSE focussed journal (*International Journal of Agent-Oriented Software Engineering*).
- The lack of true AO programming languages. Taking a historical parallel, it is often said that the object paradigm, invented in the 1960s, did not take off until languages like C++, Eiffel and Smalltalk were commercially available in the late 1980s. The perception is that current AOPLs are front-ends to Java – ergo, agents are just objects! (Such is the perception of many in both industry and academe)
- The lack of standards. Perhaps this is a little early but in the last few months both ISO and the Object Management Group (OMG) as well as FIPA have been investigating the potential increased output of standards for agents by these respective standards bodies.

I therefore, in broad terms, agree with Scott deLoach's position statement with one exception. I have always believed that it is vital to agree on *concepts* before notation. Concepts represent our shared understanding and we must therefore have an *agreed* ontology or metamodel. Only then can we ask how we might communicate those concepts within our community and to users of our work. This is the notation. Creating a notation requires different skills from creating a metamodel or ontology; for instance, it requires understanding of semiotics and usability, which are irrelevant to building a conceptual basis.

Along these lines, I have recently led an ISO study group assessing the maturity of agent technology for ISO standardization (reporting May 2008). It is likely that one of the recommendations will be to investigate the likely standardization of the conceptual basis through a metamodel – an initiative that is also being pursued within the Object Management Group commencing June 2008. Once such a conceptual base is reasonably well-established, we should then pursue a common notation.

Finally, as you might expect, I endorse Scott's comment that in 2005 I advocated – and still do advocate – situational method engineering as an excellent way to build organizational-specific and/or project-specific methodologies for use by industry. The

approach has been shown to be successful on several object-oriented projects but, to the best of my knowledge, not yet in an agent-oriented software engineering environment. This needs to be done.