TOWARDS A GENERIC METAMODEL FOR MAS WORK PRODUCTS – EXTENDED ABSTRACT

Ghassan Beydoun[a] Cesar Gonzalez-Perez[b]
Brian Henderson-Sellers[b] Graham Low[c]

[a] *Faculty of Informatics, University of Wollongong, Wollongong, Australia, beydoun@uow.edu.au,*
[b] *Faculty of Information Technology, University of Technology, Sydney, PO Box 123, Broadway, NSW 2007, Australia*
[c] *School of Information Systems, Technology and Management, University of New South Wales, Sydney, Australia, g.low@unsw.edu.au*

In the context of creating methodologies for MAS system development using a situational method engineering approach, we focus in this paper on the creation and evaluation of a generic metamodel to serve as a representational infrastructure to unify the work product component of MAS methodologies. The resultant metamodel does not focus on any class of MAS, nor does it impose any restrictions on the format of the system requirements; rather, it is an abstraction of how the work product elements in any MAS are structured and behave both at design time and run-time. Furthermore, in this paper we validate this representational infrastructure by analysing two well-known existing MAS metamodels (Islander and TAO). We sketch how they can be seen as subtypes of our generic metamodel, providing early evidence to support the use of our metamodel towards the construction of situated MAS methodologies.

This paper was originally presented at the SELMAS 2005 meeting in St Louis, MO, USA and published in LNCS 3914 [1]. It represents one of several analytical approaches to the challenge of developing a high quality modelling language (ML) for use in designing and implementing agent-oriented software-intensive systems. In an overview of current trends in European AOSE research by Bernon *et al.* [2], this need for an appropriate modelling language was underlined. Both in our papers on FAML and papers by other authors, we seek to discriminate in several dimensions:

- An agent modelling language, defined by a high quality metamodel, that describes the concepts to be used in creating an agent-oriented design. It is important, at least initially, that the concepts for modelling and documenting designs should be kept separate from the ways of doing that design i.e. we focus here on the work product aspects of a methodology and not at all on the process nor producer aspects. These are dealt with, for instance, in the new ISO/IEC 24744 forthcoming standard (2007) which stresses these aspects but also the necessary semantic linkage to the kind of "MAS metamodel" described here.
- Agents as opposed to objects. While many authors, including ourselves, see value in utilizing much of the object-oriented concepts where they are applicable to agents, there are clearly aspects of objects that do not translate well. For instance, a standard description of an OO class is that it is an entity with attributes and operations (an agent has neither) and offers services, which are executed without argument whenever they are requested by another object. Agents do not offer such "guaranteed" and blindly-followed service requests but rather take decisions based on many other factors, including environmental ones. If we take this view, then we must call into question the efficacy of creating a high quality agent modelling language on top of an object-oriented one. It is therefore a discussion point whether current efforts to extend UML into agent-oriented languages such as Agent UML [3] will provide the best solution. The arguments for such an extension tend to be market-driven rather than quality-driven.
- Run-time versus design time: in FAML we make this as a clear distinction. While there is of course overlap, practioners using, say UML in an OO environment, find it difficult to know which bits of the modelling language are relevant to which stages of the lifecycle since in UML all classes in the

metamodel are equal. In FAML, we organize our viewpoints in two dimensions: design time versus run-time and agent internals versus agent externals. Both viewpoints are discussed as areas for future research in [2].

- Genericity versus specificity: FAML aims to provide metamodel support for a high degree of abstractedness in the sense that all concepts embodied in the FAML metamodel should be usable by all methodological and design approaches, whether adaptive agent-focussed as is Adelfe [4], structurally focussed as is TAO [5] or, like Islander [6] used to describe e-institutions. The analysis of how these specific kinds of agent approaches are supported by the FAML metamodel are discussed in Section 3 of the paper.

It is important that such "dimensions" be considered seriously from a research viewpoint initially – so that a high quality ML metamodel for agents can be agreed upon by the community independently of marketing pressures. Of course, best practice should also contribute, but the industry usage of agents is significantly smaller than was the case with objects when standardization of OO MLs were being discussed a decade ago. Once agreed, then such a "lingua franca" for agents could potentially be as useful to industry adopters, tool vendors and educationalists as was UML in the OO world. It would parallel such OO initiatives (as in UML), would reuse the OO concepts and metamodel structure where appropriate but deviate when the clear distinctions between agents and objects made it necessary to do so. This paper is a contribution towards that goal.

References

[1] G. Beydoun, C. Gonzalez-Perez, B. Henderson-Sellers and G. Low. Developing and evaluating a generic metamodel for MAS work products. In A. Garcia, R. Choren, C. Lucena, P. Giorgini, T. Holvoet and A. Romanovsky, *Software Engineering for Multi-Agent Systems IV: Research Issues and Practical Applications*, 126-142, 2006.

[2] C. Bernon, M. Cossentino and J. Pavón. An overview of current trends in European AOSE research. *Informatica,* 29: 379-390, 2005.

[3] B. Bauer, J. Müller and J. Odell. Agent UML: a formalism for specifying multiagent interaction. In *Agent-Oriented Software Engineering: First International Workshop. AOSE 2000.* LNCS 1957, Springer-Verlag, 91-103, 2001.

[4] C. Bernon, M.-P. Gleizes, S. Peyruqueou and G. Picard. ADELFE, a methodology for adaptive multi-agent systems engineering. In *Engineering Societies in the Agents World,* 2002.

[5] V. Silva and C. Lucena. From a conceptual framework for agents and objects to a multi-agent system modeling language. *Autonomous Agents and Multi-Agent Systems*, 8: 1-45, 2004.

[6] M. Estava, D.d.l. Cruz and C. Sierra. ISLANDER: an electronic institutions editor. In *International Conference on Autonomous Agents & Multiagent Systems (AAMAS02),* ACM, 2002.