Modelling with Agents

Estefanía Argente\(^1\), Ghassan Beydoun\(^2\), Rubén Fuentes-Fernández\(^3\), Brian Henderson-Sellers\(^4\), and Graham Low\(^5\)

\(^1\) Universidad Politécnica de Valencia, Spain
eaargente@dsic.upv.es
\(^2\) University of Wollongong, Australia
beydoun@uow.edu.au
\(^3\) Universidad Complutense de Madrid, Spain
ruben@fdi.ucm.es
\(^4\) University of Technology Sydney, Australia
brian@it.uts.edu.au
\(^5\) University of New South Wales, Australia
G.Low@unsw.edu.au

Abstract. Modelling is gaining relevance in Agent-Oriented Software Engineering (AOSE) because of two main reasons. Firstly, the conceptual frameworks have reached a level of maturity that makes it reasonable to devote effort to seek a consensus in modelling languages, including tool support. Secondly, the influence of model-driven engineering emphasizes the potential value of having models at the core of development processes. This survey analyzes these changes in AOSE modelling languages along three dimensions. The semantic dimension refers to the concepts considered in the languages. The syntactic dimension covers the technical means by which languages are defined. The operational dimension regards the use of these languages, considering both their support and acceptance. The overall context for this discussion is the comparison of several modern AOSE approaches.

Keywords: Multi-agent system, Model, Modelling language, Comparison, Semantics, Syntax, Use.

1 Introduction

Agent-Oriented Software Engineering (AOSE)\(^{28}\) regards modelling as a key task in the development of Multi-Agent Systems (MASs). It provides a variety of supporting elements for it, such as notations, guidelines and tools. However, the use of models in AOSE has been limited by three main factors. Firstly, the paradigm has experienced difficulties to reach agreement on the concepts to be considered\(^6\). These have been continuously evolving and have frequently been ill-defined in the literature. Secondly, there has been a lack of suitable techniques to handle both modelling languages (MLs) and models\(^38\). This has resulted in conceptual frameworks frequently evolving more quickly than their formal representations, thus hampering tool support for them. Thirdly, it is hard to
capture application domains in models and consistently transform them into running code [19].

Seeking to overcome these limitations, AOSE research is working in two main directions. Firstly, the field is pursuing consensus on what the relevant concepts for modelling a MAS are as well as their definitions. This trend can be observed in the research on unification frameworks [1,6,33], mappings between notational sets [36] and toolset comparisons [15]. The result of these efforts is expected to be a widely accepted, stable and precisely defined conceptual framework for MAS modelling. Secondly, AOSE is increasingly adopting Model-Driven Engineering (MDE) [38] approaches, as it is seen in the growing number of authors who consider models to be at the core of an AOSE development process [28]. This has impact on the way we may choose to define a ML: with an increasing incorporation of metamodels and their use to generate different development artefacts; and with a growing use of automated transformations. These kinds of approaches are intended to favour a closer alignment between MLs, their support tools and the code produced from models.

In order to analyze the evolution along these two research lines, this paper considers the different aspects in the definition of a ML [29]: the semantics define the concepts present in the ML and their meaning; the syntax comprehends the abstract syntax, i.e. the representation of these concepts in terms of entities, relationships, attributes and constraints, and the concrete syntax, i.e. the actual representation of the elements of the abstract syntax, for instance with text or graphical notations. A third aspect considered here is the context in which a ML is used, including documentation, processes, support tools and diffusion.

The study of the previous aspects considers several recent and illustrative AOSE MLs. Some of them are stand-alone languages while others are an integrated part of an AOSE methodology. For convenience, we will use the term “ML” irrespective of whether the ML is published independently, e.g. FAML [6], ANote [9] and MAS-ML [39], or as part of a methodology, e.g. Adelfe [4], INGENIAS [37] and Prometheus [34,35]. The evolution and inter-relationships of all these “MLs” are depicted in Fig. 1. It can be seen that significant influences other than within the research team are few and far between. Older approaches such as Adelfe and PASSI have had some influence but the two most obvious beneficiaries of earlier work are the two projects aiming to create a convergence, FAML and the “Unifying MMM” [5]. Many methodologically-based MLs also use elements of UML [32], AUML [31] or AML [8] — three languages developed in the standards arena rather than directly in the context of a specific methodology. UML Profiles are also used, for instance in AOR’s (Agent-Object-Relationship) ML [41]. Agent modelling is also supported in i* [43], especially for the requirements engineering stage.

The MLs chosen for analysis in this survey are, in alphabetical order, Adelfe [4], FAML [6], INGENIAS [37], O-MaSE [14], PASSI [10], Prometheus [34], SODA [30] and Tropos [23] – those which, in the subjective judgement of the

---

1 Although the simple juxtaposition of several metamodels [5] without due regard to ontology unification is unlikely to be successful in the long term.