Mimosa: using ontologies for modeling and simulation

Jean-Pierre Müller, CIRAD-GREEN, Montpellier, France

1 Introduction

Modeling is the shared activity for both modeling and simulation, and knowledge representation. However knowledge representation is focused on building a description, usually static, of the reality in terms of named categories, properties, relations and objects, while the modeling and simulation community aims at putting together behavioral components [MB05] as far abstractions of the dynamical structure of the reality. To face the complexity of the systems we are trying to model and to simulate nowadays, the challenge addressed in this paper is to mix both approaches in a common framework. Trying to use modeling and simulation as a way to combine both expert and profane knowledge in support to negotiation activities[Aa03], our aim is to cover the whole modeling activity going from knowledge elicitation and representation up to running simulation models for exploring prospective options.

For doing that, we first describe separately the recent advances made in the modeling and simulation community as well as in the knowledge representation community, especially the ontologies with the arrival of the semantic web[LG04]. We show that, despite the same goal to describe a reality, or at least a part of it, it results in very different, although related concepts. From the analysis of the differences and similarities, we propose an architecture which is being tested in a modeling and simulation platform: Mimosa. The outcome is a formal way to pave the path from conceptual to running models which is sketched in this paper. The achievements and the perspectives are discussed in the conclusion.

2 Modeling and simulation

Modeling and simulation has been primarily used to model dynamical processes, initially with differential equations. In industrial contexts, the theory of control needed more sophisticated representations, essentially by composing transfer functions. This need gave rise to a number of formalisms for structuring modeling which were finally unified within a widely recognized framework called DEVS for Discrete EVent System [ZKP00]. DEVS defines a clear operational semantics and proves its closure under composition. However, the manipulation of very complex models called for even more structured modeling paradigm among which object-oriented and multi-agent systems.

Object-oriented modeling comes from the intuition that the dynamics are usually associated to devices or parts and the overall dynamics arises from the combination of these parts (e.g; mechanical devices, industrial plants, etc.). Object-oriented modeling is directly related to object-oriented languages in computer-science. However, while object-oriented languages propose the notions of *classes* and *objects* as instances of classes, only the notion of objects is provided in most modeling and simulation systems. In effect, the purpose of modeling is generally focused on the description of one system, or one category of systems to simulate. The classes of the objects constituting the system are usually predefined in the modeling and simulation platform. However, the need to combine the components in complex simulations called for defining the semantics of these components as proposed in [BMdC$^+$].

Multi-agent modeling extends the object-oriented paradigm in three ways: i) the organization of the interactions between objects can change over time (usually by having the objects moving within a topological space), ii) the objects can dynamically appear and disappear, iii) the dynamics of the objects can be anything from transfer functions to sophisticated reasoning, passing by rule-based systems. These objects are called *agents*. The natural heterogeneity of the agents of a multi-agent system calls for the description of classes of agents which can be instantiated into populations in various ways, including dynamically during the simulation. The drawbacks of this generality is that there is hardly any formal language nor any clearly defined operational semantics for modeling with multi-agent systems. Nevertheless, it exists some attempts to formalize, either the dynamical interaction structure of the agents using extensions of DEVS[Uhr03] or some agent architectures.

3 Knowledge representation and the ontologies

Independently of the modeling and simulation community, Artificial Intelligence (AI) developed the domain of knowledge representation in order to investigate how human beings talk and reason about the reality. Initially based on formal logics, AI moved rapidly towards more structured representations like frames[Min75] or conceptual graphs[Sow98], generally called *object-centered representations*. In these representations, one also distinguishes between *concepts* (describing categories of objects by their shared structures) and *instances* as describing the objects in their unicity. The concepts and instances are described by their attributes and relations with, respectively, other concepts or instances. The notion of generalization is defined between concepts: e.g. the concept of animal is considered more general than the concept of elephant. The notions of concept and instance in object-centered representations does not entirely fit the notions of class and object in object-oriented languages. For example, in knowledge representation, an instance is a kind of concept which denotes (describes) an individual while in object-oriented programming, an instance *is* an individual. As a consequence, in object-oriented programming, an instance has a state which evolves independently of the description of the related class. It is not the case in knowledge representation where the notion of state is meaningless because one only describes what is always true about concepts and instances.

The last outcome of the knowledge representation domain, with the advent of the semantic web, is the notion of ontology. According to Tom Gruber at Stanford University[Gru93], the meaning of ontology in the context of computer science, however, is "a description of the concepts and relationships that can exist for an agent or a community of agents.". Contemporary ontologies share many structural similarities, regardless of the language in which they are expressed. Most ontologies describe individuals (instances), classes (concepts), attributes, and relations including the generalization.

4 Mixing both: the challenges

If ontologies seem appropriate to describe the world and, then, to provide means for describing categories of objects and systems, they were only recently used for modeling with the aim of building models for simulation for mainly two reasons: the first reason is technical: the ontologies are mostly used to describe the concepts we are talking about, providing the so-called *conceptual models*. The path from conceptual models to simulation is a multi-step process. One way is to annotate existing simulation components with ontological descriptions [BMdC$^+$]. The second reason is semantical: as mentioned earlier when comparing knowledge representation and object-oriented programming, the notion of instance does not correspond to the notion of object. For simulation an object has a state because the system evolves, for knowledge representation an instance does not have any state because the whole representation is made for reasoning about a state of the reality, not for evolving it.

Despite existing work on using ontologies for modeling and simulation, either by providing an ontology of the modeling process itself like in the FEARLUS project[PEP$^+$05] or an ontology of simulation formalisms and component types like in DeMO[MB05], the integration from conceptual modeling to simulation still needs to be done with two main challenges: a) semantically, to understand and to formalize the relationship between ontologies and model implementations, in particular between conceptual models and models. b) technically, to use this formal account for automate this process as proposed in [dLV02].

5 Our proposal

Mimosa[1] is an extensible modeling and simulation platform ([MÖ4]) used to investigate the above mentioned challenges.

A *conceptual model editor* allows the definition of ontologies using a subset of the UML class diagram equivalent to ontology languages like OWL[owl04] or others (i.e. restricted to properties, generalization and relations). The user can define the concepts, their attributes and their relationships. Because the ontologies must define what is universally true, only information like the cardinality of the attributes and relations, the types of the attributes and the concepts a concept can be related to, are described.

[1] It is the french acronym for "Méthodes Informatiques de MOdélisation et Simulation Agents": computer science methods for agent-based modeling and simulation

A separated dynamical description can be attached to each concept describing the state (distinct from the attributes) and its dynamics using any formalism which can be mapped into DEVS. This is the proposed way to combine structural and behavioral descriptions.

A *model editor* manages the instantiation of the conceptual model into a model, using a superset of the UML object diagram. This model is considered as the description of the reality at time 0. In particular, we must define the actual value of each attribute as well as the actual relations between the objects in the initial situation. Being still at the knowledge representation level, we insist that the object diagram is a description of the reality at the initial time made of instances, with two consequences:

1. any modification made on the conceptual model is reflected in the model itself, including adding or removing attributes and relations.

2. it opens the path to build the conceptual model from the model (using instances as templates for concepts).

Finally, a running model is created by generating the objects in the object-oriented programming sense from the instances, creating the states which are initialized from the descriptions and which shall evolve by simulation. The *scheduler* is in charge of generating the initial state of the system before running the model.

6 Conclusion

Most of what is described in the previous section has been implemented and is downloadable from http://sourceforge.net/projects/mimosa. There are two directions remaining to explore: i) a closer interoperability with ontological languages. It relies on the possibility to actually load ontologies made with other systems like Protegee as a base for building simulation models. In the other hand, the possibility to export our conceptual models towards ontology-based systems would expand the possibility to manipulate descriptions and models using XML, as well as adding reasoning capabilities; ii) the Mimosa platform is oriented towards multi-agent modeling, Therefore, an extension of DEVS towards dynamical organizations as in [Bar98, SW06] has to be defined in order to be able to map various ways of describing multi-agent systems into a well defined operational semantics.

Finally, the need to integrate the multiplicity of points of view and scales of description for describing complex systems calls for departing from ontologies in the strict sense (i.e. discourse about beings) to go towards "epistemologies": i.e. discourse about points of view on beings, and ways to articulate them. The AGR[FG98] paradigm within the multi-agent system community is going in this direction.

References

[Aa03] Martine Antona and al. Our companion modelling approach. *Journal of Artificial Societies and Social Simulation*, 6(2), 2003.

[Bar98] Fernando J. Barros. Abstract simulators for the DSDE formalism. In D.J. Medeiros, E.F. Watson, J.S. Carson, and M.S. Manivanan, editors, *Proceedings of the 1998 Winter Simulation Conference*, 1998.

[BMdC$^+$] David Bell, Navonil Mustafee, Sergio de Cesare, Mark Lycett, and Simon J.E. Taylor. A Web Services Component Discovery and Deployment Architetcture for Simulation Model reuse.

[dLV02] Juan de Lara and Hans Vangheluwe. AToM: a Tool for Multi-Formalism and Meta-Modelling. In *Fifth Fundamental Approaches to Software Engineering*, LNCS. Sprenger, 2002.

[FG98] Jacques Ferber and Olivier Gutknecht. A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. In *Proceedings ICMAS '98*, 1998.

[Gru93] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.

[LG04] Lee Lacy and William Gerber. Potential modeling and simulation applications of the web ontology language - owl. In R.G. Ingalis, M.D. Rossetti, J.S. Smith, and B.A. Peters, editors, *Proceedings of the 2004 Winter Simulation Conference*, 2004.

[MÖ4] Jean-Pierre Müller. The Mimosa generic modeling and simulatiion platform: the case of multi-agent systems. In Herder Coelho and Bernard Espinasse, editors, *5th Workshop on Agent-Based Simulation*, pages 77–86, Lisbon, Portugal, May 2004. SCS.

[MB05] John A. Miller and Gregory Baramidze. Simulation and the semantic web. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J.A. Joines, editors, *Proceedings of the 2005 Winter Simulation Conference*, 2005.

[Min75] Marvin Minsky. A framework for representing knowledge. In P. Winston, editor, *The psychology of computer vision*. McGraw-Hill, 1975.

[owl04] http://www.w3.org/2004/OWL/, 2004.

[PEP$^+$05] E; Pignotti, P. Edwards, A. Preece, G. Polhill, and N. Gotts. Providing Ontology Support for Social Simulation. In *Proceedings of the First Internatonal Conference on eSocial Science*, Manchester, 2005.

[Sow98] John F. Sowa. Conceptual graphs. In P. Bernus, K. Mertins, and G. Schmidt, editors, *Handbook on Architectures of Information Systems*, pages 287–311. Springer Verlag, 1998.

[SW06] Hui Shang and Gabriel Wainer. A Simulation Algorithm for Dynamic Structure DEVS Modeling. In L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, and R.M. Fujimoto, editors, *Proceedings of the 2006 Winter Simulation Conference*, 2006.

[Uhr03] Alinde M. Uhrmacher. Simulation for Agent-Oriented Software Engineering. In W. H. Lunceford and E. Page, editors, *First International Conference on Grand Challenges*, San Diago, California, 2003.

[ZKP00] Bernard P. Zeigler, Tag Gon Kim, and Herbert Praehofer. *Theory of Modeling and Simulation*. Academic Press, 2000.