

MODELING OF SOCIAL SYSTEMS WITH INGENIAS

Juan Pavón, Candelaria Sansores, Jorge Gómez-Sanz

*Universidad Complutense Madrid, Facultad de Informática, Madrid 28040, Spain,
jpavon@pdi.ucm.es, csansores@fdi.ucm.es, jjgomez@pdi.ucm.es*

Paper published at Workshop on Multi-Agent Systems and Simulation (MAS&S), Proc. of the 4th Int. Industrial Simulation Conference (ISC'2006) June 5-7, 2006, Palermo, Italy, pp. 9-16

Extended Abstract

There are good toolkits for agent based simulation (ABS). However, for many potential users of ABS, such as sociologists or economists, it is still very difficult to specify models with such tools. This is mainly because this task requires writing code in some, usually object-oriented, programming language. There are some attempts to facilitate the specification with graphical tools and a set of predefined behaviours. For instance, Sesam (www.simsesam.de/) provides a graphical interface to specify agent behaviours as a kind of state machines, and there are also some rapid application development environments, for instance using the Python language in RePast Py (repast.sourceforge.net/repastpy/). But these are only applicable to simple models. Another issue is that the agent models initially supported by the different toolkits are rather simple, mostly reactive agents, whose behaviour is specified programmatically.

On the other hand, there are many proposals of rich agent modeling languages in the agent-oriented software engineering domain. These languages are usually higher level, based on some graphical notation, and, in some cases, easily customizable. Their capabilities make them more suitable for depicting models than programming languages. Also, the supported agent model is richer, both at micro (agent) and macro (organization) levels, than in the agent based simulation toolkits. Furthermore, agent modeling concepts are closer to the abstractions that a sociologist or economist could use to model social systems. For these reasons, we consider that these agent-oriented languages are more appropriate for modeling social systems as multi-agent systems (MAS) than plain program code.

One way to perform the simulation of an MAS model would be to directly execute the specification, but this requires implementing simulation facilities adapted to the specific modeling language. Other possibility is to transform the MAS model to code for some simulation toolkit that could execute it. The execution means animating the specification, and being able to get some information on how the system evolves over time. This is not normally supported by agent-oriented software engineering tools. But agent-based simulation toolkits do this very well. So we are considering the use of an agent-oriented modeling language to specify MAS models representing complex social systems, and automatic generation of code on simulation platforms by applying transformations from the MAS models. This idea of transformation is in line with model driven engineering (MDE) practices.

To implement this approach we require an agent oriented methodology with (1) a suitable language to specify complex social systems as MAS and (2) tools to support the transformation between MAS models and simulation code. Both requirements are satisfied by INGENIAS [3], and this is the main reason for selecting it in this work. This methodology is supported by a set of tools, the INGENIAS Development Kit (IDK, available at ingenias.sourceforge.net), which facilitates the edition of models and the definition of transformations for automatic code generation. The foundation of INGENIAS is the definition of multi-agent system (MAS) meta-models, which can be customized to particular application domains. This is interesting if we need to extend the existing notation to cope with new issues that may be required for the specification of a social system simulation model.

The paper presents how INGENIAS modeling language can be used to specify social systems and how transformations can be built to generate code on RePast (repast.sourceforge.net) [2]. This platform has been chosen as it is one of the most well-known and supported agent based simulation toolkits, and its principles are similar to others such as Swarm and Mason, so the solution could be easily extended to those. The next section discusses about what elements and concepts should be considered in a language for modeling of social systems. This is followed by a section where the INGENIAS modeling language is reviewed to see how it supports the requirements for social systems modeling, and how it can be extended [5]. Then, a section describes how to make the transformation from INGENIAS social system specifications to RePast models. These models can be used for simulations on the RePast framework. This has been already experimented for RePast and Mason [4], in

order to see the consequences of using different agent based simulation toolkits. The conclusions summarizes some of these results and discusses the limitations and perspective of this approach.

The mapping from INGENIAS models to RePast is implemented by an IDK module. The module has to traverse specifications, extract information from specifications, and put the extracted information into templates of RePast agents. Environment constraints (such as spatial and temporal aspects) can also map to the main program.

The IDK module for RePast code generation is developed as an iterative process through several steps. This process can be applied similarly for other agent based simulation toolkits. The basis for code generation is the availability of code templates for the target platform, RePast in this case (see Figure). This is usually the most difficult to obtain as it requires a good knowledge of how to implement agents in the target platform. Our experience has shown that this can be accomplished through an iterative process, in which the developer defines progressively the architecture of the code for the target platform and the transformations from specification to code templates.

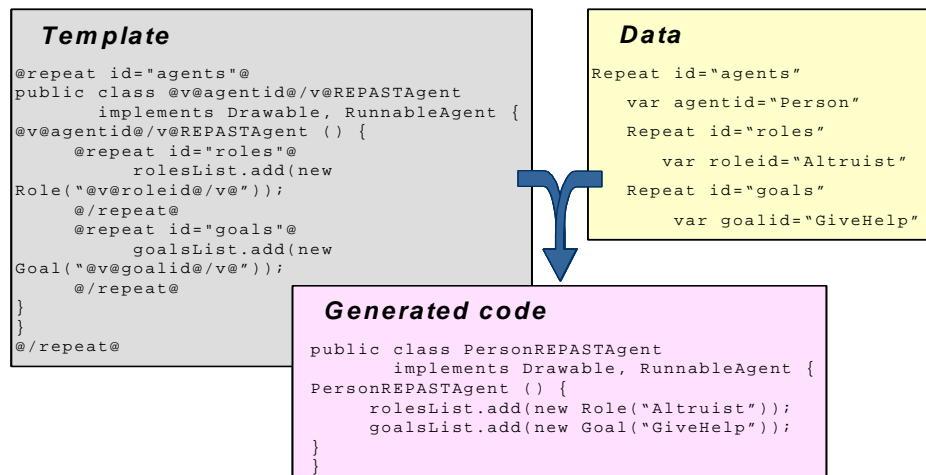


Figure 1: A simplified example of code generation

An interesting issue to consider for further work is the ability to extend the INGENIAS MAS meta-model to create domain oriented specification languages. This would facilitate even further the modelling activity of social scientists as they could use concrete domain concepts rather than pure agent concepts from INGENIAS. This work has to be done with domain experts (sociologists) and this is a task that we are currently addressing.

Acknowledgements

This work has been developed with support of Dirección General de Universidades e Investigación de la Consejería de Educación de la Comunidad de Madrid (Spain) and Universidad Complutense de Madrid (Grupo de investigación consolidado 910494 and the project TIN2005-08501-C03-01, funded by the Spanish Council for Science and Technology).

References

- [1] Castelfranchi, C. 1998. "Modelling social action for AI agents." *Artificial Intelligence* 103 (1-2), 157.
- [2] Collier, N., T. Howe, M. North (2003). Onward and Upward: The Transition to Repast 2.0. In: Proceedings of the First Annual North American Association for Computational Social and Organizational Science Conference. Electronic Proceedings. Pittsburgh, PA USA (June 2003),
- [3] Pavón, J., Gómez-Sanz, J.J. and Fuentes, R. 2005. "The INGENIAS Methodology and Tools". In: Henderson-Sellers, B. and Giorgini, P., editors: *Agent-Oriented Methodologies*. Idea Group Publishing, 236—276
- [4] Sansores, C. and Pavón, J. 2005. "Agent-Based Simulation Replication: A Model Driven Architecture Approach". In: *4th Mexican International Conference on Artificial Intelligence (MICA I 2005)*. Lecture Notes in Artificial Intelligence, Vol. 3789. Springer-Verlag, 244-253
- [5] Sansores, C., Pavón, J. and Gómez-Sanz. 2006. "Visual Modeling for Complex Agent-Based Simulation Systems". In: J.S. Sichman and L. Antunes (Eds.): *Int. Workshop on Multi-Agent-Based Simulation 2005*, Lecture Notes in Artificial Intelligence, Vol. 3891, Springer-Verlag, 174–189