Peer-Olaf Siebers and Paul Davidsson (2015)

Engineering Agent-Based Social Simulations: An Introduction

Journal of Artificial Societies and Social Simulation 18 (3) 13
<http://jasss.soc.surrey.ac.uk/18/3/13.html>

Received: 02-Apr-2015 Accepted: 10-Apr-2015 Published: 30-Jun-2015

Abstract

This special section on "Engineering Agent-Based Social Simulations" aims to represent the current state of the art in using Software Engineering (SE) methods in ABSS. It includes a mixture of theoretically oriented papers that describe frameworks, notations and methods adapted from SE and practice-oriented papers that demonstrate the application of SE methods in real world ABSS projects.

Keywords:
Agent-Based Social Simulation, Software Engineering, Software Architectures, UML

1.1 Engineering methods are still not commonly used when developing Agent-Based Social Simulations (ABSS). There was some cross-disciplinary work in the early 2000s to apply Software Engineering (SE) methods in ABSS and vice versa (e.g. Parunak & Odell 2001; Davidsson 2001). An example of such efforts was the development of AgentUML which allowed considering social aspects (e.g. roles) when defining software agents (Bauer et al. 2001). Also Agent Oriented Software Engineering research started as an attempt to make agent-oriented methodologies (e.g. Gaia or Tropos) interoperable (Weiß 2001; Bergenti et al. 2004). Some researchers demonstrated how to use SE methods in ABSS, but cross-disciplinarity was more strongly oriented towards using Social Science within Engineering. More recently, efforts to establish SE methods in ABSS have been picked up again by several researchers (e.g. Bersini 2012). One step in this direction is the application of UML and AgentUML in ABSS. Moreover AI decision-making models have been applied in ABSS (e.g. BDI in crowd simulations). In addition some tools have emerged that support the definition of ABSS models using UML (e.g. AnyLogic and Repast Simphony).

1.2 How ABSS and SE can be linked is demonstrated by the gaming industry where agent state machines in combination with social theories are used for modelling sociable characters (as described in Mac Namee & Cunningham 2001). An example for such actors can be found in The SIMS™ (Maxis 2015) – an interactive organisational agent-based simulation game. However, these game agents and the notation used to describe them during the development process are not often used for research applications in ABSS. SE also provides some project management strategies that can be useful for developing ABSSs, as for example Scrum (García-Magariño et al. 2009). Agile methods such as Extreme Programming (XP), with short development cycles, pair programming, and test driven development, also seem to have potential (Nastar & Wallman 2009). SE can be used at different stages of the development process: for management, analysis, modelling and implementation. Nevertheless, it is important to realise that many methods or processes are not transferrable one-to-one and require some adaptation to be useful within the ABSS context.

1.3 With this special section we aim to represent of the current state of the art in using SE methods in ABSS. Once we posted the call for paper we received responses confirming that this is a long needed contribution to the Agent-Based Modelling (ABM) community. The result is a mixture of theoretically oriented papers that describe frameworks, notations and methods adapted from SE and practice-oriented papers that demonstrate the application of SE methods in real world ABSS projects.

1.4 We received a relatively small number of submissions (11) indicating that this is still an area where more research is needed to be conducted. This is also the joint view that is expressed in all papers that form this special section. We hope that the collection of high quality papers in this special section will allow others to take a methodological approach towards employing SE methods for the development of ABSS.

1.5 The special section contains five papers. The first paper by Collins et al. focusses on standardisation of processes. It provides the finding from a series of workshops that were organised by the team of authors to discuss the topic of standardisation of ABM

in general. It is concluded that the ABM community should both adopt existing standards that are relevant and exploit the already existing standards processes and organizations to develop new ones.

1.6 There are three papers that focus on methods and tools. The first of these by Scherer et al. focuses on agent-based policy modelling and provides a conceptual model-driven approach developed and implemented in the OCOPOMO project. The approach ensures traceability by integrating technologies for agent-based social simulation, semantic web and model-driven development. The second paper in this block is by Rossiter and identifies a gap in the simulation literature regarding the overall structural design of simulations, and what architectures and features are valuable for what reasons. For solving this issue the author proposes to employ key software properties which embody SE best-practices. The paper provides an 'idealised' software architecture for simulation – similar to a reference architecture in SE. The application of such a reference architecture is demonstrated by a multi-paradigm model of health and social care. The third paper on methods and tools by Ozik et al. describes the new statecharts framework that was introduced in the latest release of the simulation tool Repast Simphony and provides some use cases to demonstrate how it can be used to easily design and build complex state-based agent models.

1.7 The final paper of the special section by Moyo et al. focusses on the application of SE methods to ABSS. It presents a case study where SE methods have been applied to a real world research problem. It discusses the experience of using Scrum, an agile software development method, to organise the workflow and guide the development of an agent-based model of alcohol consumption.

Acknowledgements

We would like to thank the numerous (very patient) reviewers for their valuable support.

References

BAUER, B., Müller, J. P., & Odell, J. (2001). Agent UML: A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering, 11*(03), 207–230. [doi:10.1142/S0218194001000517]

BERGENTI, F., Gleizes, M. P., & Zambonelli, F. (Eds.). (2004). *Methodologies and Software Engineering for Agent Systems: The agent-oriented software engineering handbook,* (Multiagent Systems, Artificial Societies, and Simulated Organizations, Vol. 11). Springer Science & Business Media. [doi:10.1007/b116049]

BERSINI, H. (2012). UML for ABM. *Journal of Artificial Societies and Social Simulation, 15*(1), 9: http://jasss.soc.surrey.ac.uk/15/1/9.html

DAVIDSSON, P. (2001). 'Multi agent based simulation: beyond social simulation.' In *Multi-Agent-Based Simulation*, Lecture Notes in Computer Science, Vol. 1979 (pp. 97–107). Springer Berlin Heidelberg.

GARCÍA-MAGARIÑO, I., Gómez-Rodríguez, A., Gómez-Sanz, J., & González-Moreno, J. C. (2009, January). 'Ingenias-Scrum development process for multi-agent development.' In International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008), *Advances in Soft Computing*, Vol. 50, (pp. 108–117). Springer Berlin Heidelberg.

MAC NAMEE, B., & Cunningham, P. (2001). *A Proposal for an Agent Architecture for Proactive Persistent Non Player Characters*. Trinity College Dublin, Department of Computer Science.

MAXIS (2015). http://www.thesims.com/

NASTAR, M. & Wallman, P.(2009). 'An interdisciplinary approach to resolving conflict in the water domain.' In *Information Technologies in Environmental Engineering* (pp. 411–424). Springer Berlin Heidelberg. [doi:10.1007/978-3-540-88351-7_31]

PARUNAK, H. V. D., & Odell, J. J. (2002). 'Representing social structures in UML.' In *Agent-Oriented Software Engineering II*, Lecture Notes in Computer Science, Vol. 2222 (pp. 1–16). Springer Berlin Heidelberg.

WEIß, G. (2001). Agent orientation in software engineering. *The Knowledge Engineering Review, 16*(04), 349–373. [doi:10.1017/S026988890100025X]