

# PET – A PARTICIPATORY EXTENSION FOR AGENT-BASED SIMULATIONS

Laszlo Gulyas<sup>(a,b)</sup>, Vilmos Kozma<sup>(a)</sup>, Robert Szalai<sup>(a)</sup>

<sup>(a)</sup>AITIA International Inc, Budapest, Hungary

<sup>(b)</sup>Lorand Eotvos University, Budapest, Hungary

[lgulyas@aitia.ai](mailto:lgulyas@aitia.ai), [vkozma@aitia.ai](mailto:vkozma@aitia.ai), [rszalai@aitia.ai](mailto:rszalai@aitia.ai)

## ABSTRACT

Agent-based modeling and simulation (ABM) is a powerful means to understand collective behavior as a result of individual actions. It is a new branch of computer simulation finding its use mainly in computational social science. It is well suited to take into account the heterogeneity of individuals, limited cognitive abilities and unique interactions. ABM builds the model from 'the bottom-up', individual behaviors and looking at the emergence of macro behavior. According to Epstein and Axtell, in these *artificial societies*, social structure and group behavior emerges from the interaction of individuals, under rules that place only bounded demands on each agent's information and computational capacity.

Participatory simulation, on the other hand, is an offspring of agent-based modeling that provides means for artificial agent - human interactions. In participatory models some or all agents can be controlled by individuals. These simulations are powerful educational and training tools, on the one hand, while they also provide means to blend computational simulations with the laboratory experiments of, e.g., behavioral economics.

The Participatory Extension (PET) is a tool to web-enable agent-based models and to optionally transform them to participatory simulations, where human subjects control some of the agents via the world-wide web. PET is a multi-user environment, based on a dedicated network server. It is intended as an educational and/or laboratory tool. Therefore, it has two distinct user roles. The first is that of the 'teacher' or 'lab director' who can set up the environment for the 'students' or the 'subjects'. The teacher determines what *model families* (i.e., simulation programs) are available on the PET server, she configures the *models* available to the students (i.e., with what specific agent composition, with what parameter values fixed and what values to be set by the students, etc.). The students, on the other hand, enter via a simpler user interface that lists the pre-configured models available to them. Depending on the settings by the teacher, students may be able to start their own *simulation copies*, or can join to watch the progress of the simulation initiated by someone else (e.g., by the teacher).

The teacher/lab director may also decide to make some agents in the model 'controllable', which means that students (subjects) joining the simulation can take control over them. Students that control agents are provided with a special graphical user interface that conveys the information available to the agent and provides 'actions' by which they can specify the agent's behavior. (A rudimentary control interface, in which actions are listed in the form of web links, is automatically generated by PET for any model. More advanced interfaces can be provided by Java programming.) In PET all simulations are recorded and are available for future playback, including those containing human controlled agents.

PET is part of the Multi-Agent Simulation Suite (MASS, <http://mass.aitia.ai/>) that works with several 'simulation cores'. Simulations for PET can be written in its Java-based simulation platform called Multi-Agent Core (MAC). Simulations created using the popular agent-based simulation environment Repast J (<http://repast.sourceforge.net/>) can also be automatically uploaded to the PET server – even models written without PET in mind.

This paper introduces the capabilities of PET using working examples and discusses the various related concepts and technical issues.

## ACKNOWLEDGMENTS

The work reported here benefited from a grant from the Hungarian Government (grant #GVOP-3.2.2-2004.07-005/3.0). The partial support of the European Commission via the FP6 STREP projects QosCosGrid (contract #033883) and EMIL (contract #033841) is also gratefully acknowledged. László Gulyás also acknowledges the support of the Hungarian Government (OTKA T62455).