MODEL EXPLORATION MODULE

[*]M.D. IVÁNYI, ELTE-IKKK SIMULATION CENTER, BUDAPEST, HUNGARY
L. GULYÁS, AITIA INTERNATIONAL, INC., BUDAPEST, HUNGARY
R. BOCSI, AITIA INTERNATIONAL, INC., BUDAPEST, HUNGARY
G. SZEMES, ELTE-IKKK SIMULATION CENTER, BUDAPEST, HUNGARY
R. MÉSZÁROS, AITIA INTERNATIONAL, INC., BUDAPEST, HUNGARY

INTRODUCTION

MEME stands for the Model Exploration Module of the Multi-Agent Simulation Suite developed at AITIA International, Inc. MEME is an agent-based simulation tool enabling modelers to design experiments, run models on clusters and grids of computers, and to manage, analyze and visualize the data produced. Development of the tool begun in mid 2006 after the realization that there was hardly any software available being able to run a large number of simulations, collect, organize and visualize their data for modelers with limited programming skills. This paper gives an insight into the motivations of the development and introduces the reader to the tasks and functions MEME is capable of.

Keywords: Agent-Based Modeling, Modeling Tools, Simulation, Experiment Design

MOTIVATIONS

Agent-based modeling and simulation based computational science demonstrates great promise, but as Bankes and Lampert (2004) argues, so far it has lacked the rigor that is needed in the scientific field and the robustness required in policy making. Models of complex social systems typically depend on a number of assumptions, quantified in the form of specific values to certain model parameters. Ideally, any such model should be tested with any meaningful combination of these parameters, in order to determine the validity of the model or ensembles of models. This process is often called the *parameter space search* or *parameter sweep*. Additionally the task of establishing the results' statistical validity also involves running the simulation with various random number generator seeds and analyzing the collected results.

[*] *Corresponding author address:* Marton D. Ivanyi, ELTE-IKKK Simulation Center, Pazmany Peter Setany 1/C, Budapest, Hungary, H-1117; e-mail: mivanyi@aitia.ai

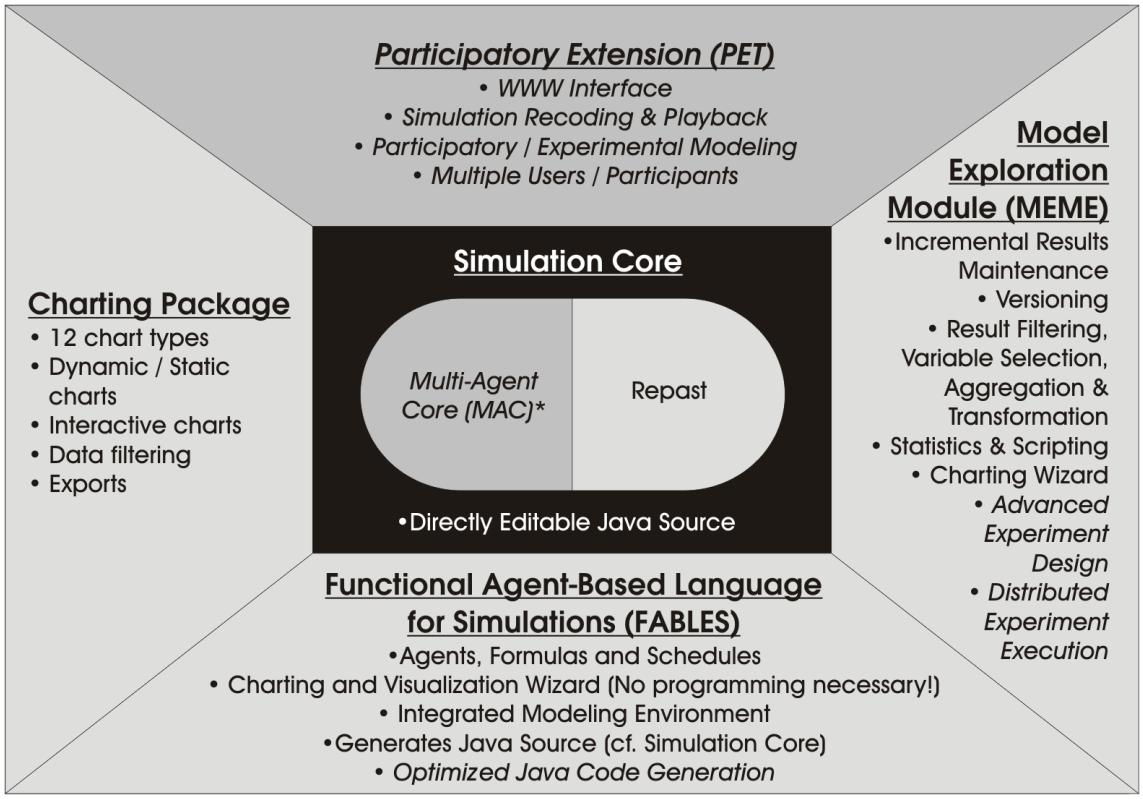

FIGURE 1 MEME is just an element of the Multi-Agent Simulation Suite (structure diagram above).

Running simulations at this scale and complexity is from a computational point of view a particularly intensive task. It is understood that it is essentially impossible to achieve complete parameter space exploration. This creates a special emphasis on the design and sequencing of the experiments, so that they allow for branching depending on earlier results and also for revisiting previously explored areas with greater 'resolution'. Even in case of carefully designed experiment plans, these tasks may exceed the abilities of today's PCs or workstations. Therefore it would be desirable to distribute simulations among several computers on the network, on a local cluster or a grid.

AVAILABLE APPLICATIONS

Standard ABM modeling packages like Swarm, Repast or NetLogo, all offer some support for collecting results in 'batch mode', as opposed to the primary, 'GUI mode' running of simulations. Swarm and Repast require (and support) modelers to create a distinct, 'batch mode' version of their models, offer a 'parameter language' to describe the regions of the parameter space the user wishes to explore, and collect the results in a specially formatted text file. The expressiveness of the offered parameter languages, however, limits the search strategies mostly to regular hyper-parallelepipeds. In particular, no dynamic branching of the exploration is possible, except by hand (i.e. processing the results of one batch and writing the next parameter file appropriately). Furthermore Swarm doesn't have a graphical user interface for designing

parameter files, while the latest version of Repast has a simple GUI for parameter space explorations.

NetLogo, on the other hand, provides a GUI for model exploration, but no parameter file. None the less, the functionality of the tool is about the same as with Swarm and Repast. A significant difference is that NetLogo does not require the user the create a dedicated batch version of the model – at the price that by default NetLogo animates all displays and graphs during parameter space exploration, thus slowing down its execution. Lately turning off displays and graphs during batch runs has been introduced in NetLogo.

General-purpose, modeling package independent parameter space exploration toolkits also exist, like Drone and SweepOver. However, they also share most of the problems discussed above: they are typically highly technical, providing no easy-to-use user interface, but requiring a kind of programming. Moreover, they are suitable for exploring the regular regions of the parameter space (i.e. hyper-cubes and parallelepipeds), but not for adaptive branching, etc.

MODEL EXPLORATION MODULE

With the Model Exploration Module we want to provide a tool that makes the parameter space exploration experience as smooth and trouble-free, in other words user-friendly, as possible. In particular, modelers should be users and not programmers of this tool, who focus on the *modeling problem* and not on the technicalities. The goal with the MEME development is to provide easy means for the common parameter sweep tasks, while supporting model specific, complex task in addition. MEME offers easy-to-use graphical user interfaces for all its basic functions. The modeler can set up and run experiments from regular ABM models, execute batch runs on clusters of computers, collect, organize, import, visualize and do basic statistics from data through the GUI without having to write a line of code.

On the other hand the software offers advanced users optional scripting when setting up the parameter sweeps or importing and organizing results for example. Scripting requires more technical expertise from the user of course, as mentioned before it is optional, and was introduced for modelers whose needs exceed MEME's rich functionality.

The design and development of the tool is done in a modular way, with the components communicating through well defined APIs, allowing for the seamless integration of new or improved functionality. We have just recently started the Beta testing of a processing plug-in that will allow the user to run simulations on local clusters of computers and a number of existing grid solutions. We hope that this development will further reduce execution times, hence speed modeling work up without requiring the modeler to develop any special grid knowledge or skills.

Experiment Design

Agent based simulation is usually related to studying the behavior of certain real phenomena. In order to do scientific experiments various factors of long and large number of

unattended simulations have to be observed. The simulation runs are executed on parameter spaces that are pre-defined or better yet, as discussed latter on, intelligently identified during the runs.

MEME currently offers solutions for designing experiments from models written in the popular [1]Repast environment and FABLES – a simple modeling language and its integrated modeling environment also by AITIA, see Gulyas and Bartha (2005) – without any additional programming. MEME offers a graphical user interface that fully assists the modeler in the process of setting up the experiments in a user-friendly way.

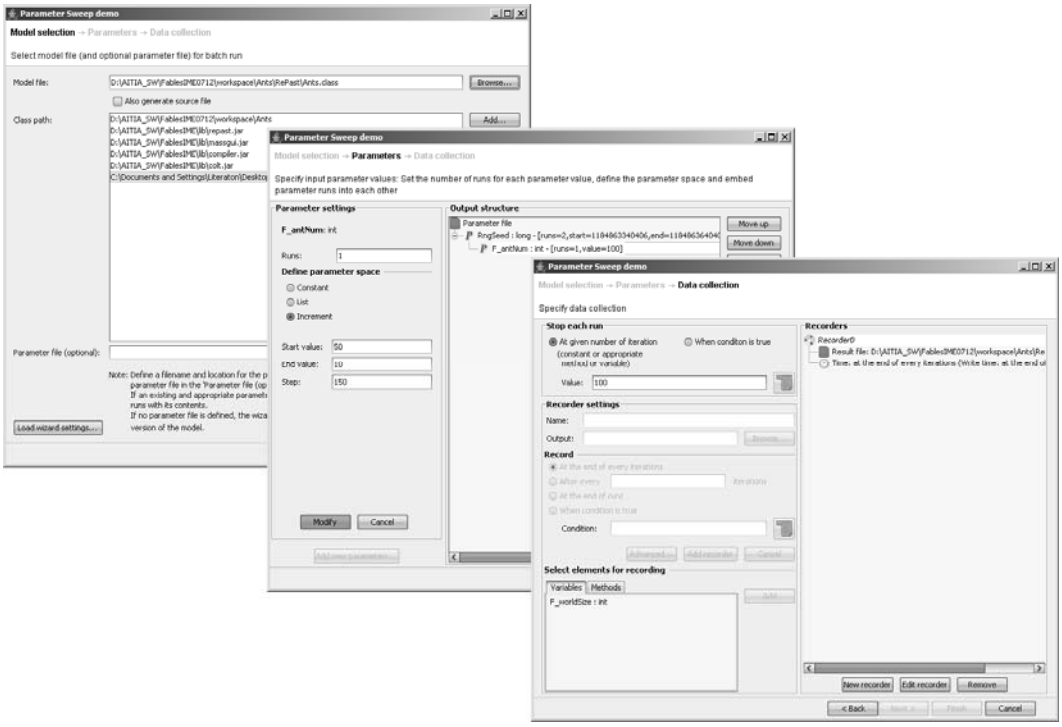

FIGURE 2 The parameter sweep wizard: selecting the model and required packages, setting up the parameter space and the data recording attributes.

Regular (i.e. not explicitly batch version) Repast and FABLES models can be imported into the application through a wizard that automatically identifies whether and which additional classes and jars that are needed to run the simulations. The same wizard explores the model for input and output parameters, lets the user create additional parameters if desired and provides the interface for declaring a parameter space for the experiment. The interface also offers capabilities to pick/add the variables whose values will be recorded and allows the user to define other data recording attributes. Measurements can be done at the end of simulations or at any desired tick count and in all cases measurements can record the actual value of the given variable, or the result of simple statistic operations. For advanced users MEME also offers scripting for declaring

[1] The tool is also developed to be able to deal with other Java-based modeling languages in the future.

entirely new variables that are results of various operations done on one or more of the original measurements.

Upon completing these tasks a new model and the corresponding parameter file are automatically generated from the original model with all the additional code necessary for running the simulation with the desired settings either on a local machine or to be distributed set as a set of simulation runs on a pool of available hosting computers.

Experiment Execution

The models and parameter files resulting from execution are forwarded to the respective engines for running Repast and Fables models in batch mode. In this phase of development, only 'brute-force' exploration of regular subspaces of the parameter space is supported. The engines are capable to run in a detachable, 'head-less' fashion. That is, after the GUI wizards have set up the exploration schemes, the engines require no user interactions and are capable to run in the background as separate applications, while providing means for the user to observe the incoming results and oversee the general state of the experiment while executing a high number of simulations.

Depending on the projected processing time of the batch runs the user has the option of launching the simulations on a single computer, a local or a remote cluster of computers. The module that distributes the simulation runs on the available computers was developed with extreme care in regard to collecting and storing the results of distributed runs. The module is prepared to re-instantiate runs to different computer once a host becomes unavailable or corrupted results are returned. As various systems with similar (albeit not simulation tailored) capabilities are already available, we did not intend to develop our own grid solution. To the contrary, we have and are developing a grid module that is capable of running simulations on an extendable list of distributed environments.

MEME currently is able to set up and run simulations with Inria's open source ProActive solution *(Caromel et al. 2006)* and the market leading Platform LSF family. These solutions enable the modeler to execute sets of simulation runs on clusters of computers running on [2]Windows or Linux/Unix operating systems, using both shared and separated storage.

Storing, Organizing and Visualizing Simulation Data

The MEME stores simulation results in a database, that includes all fix (constant) and changing parameters, and various additional information about the model (i.e. name, version, description, etc.). The software has a built-in Java-based database engine that can manage databases up to 8 GB in size, but the system is built in a way that it is independent of the

[2] Note, that ProActive is an SSH-based grid and as Windows platforms do not have a general SSH solution, preparing a Windows cluster for MEME carries some additional challenges.

particular SQL engine used, it supports professional database engines through the JDBC protocol. The program organizes raw data in a 3-level hierarchy (model, version and batch), the database-structure is created to be able to handle repeated exploratory runs, iterative, gradual import of results (i.e. new parameters being introduced and old ones deleted between versions of the same model).

MEME can obtain simulation results in two general forms. Results from batches of simulations designed and ran through MEME are automatically acquired. The other option is running simulations separately and then importing Repast result files or CSV files into the database. The program supports multiple file import, organizing data into different versions or batches under the same model. MEME distinguishes between input (i.e. values that do not change in during a run) and output parameters in result files. Various import settings can be saved if needed for later use, when importing data from a high number of simulations.

An important feature of model exploration is the processing of the available results. From the results obtained and stored in the database, subsets can be created. These computed tables, that we call ‘views’, can be described as tables, where variables (including parameters and measured variables) are in the columns and particular value-combinations are in the individual rows. Creating basic views for visualization and/or to be imported into other applications for more sophisticated analysis includes selection of the variables to be included from the results, plus specifying what model version(s) the variables be originated from.

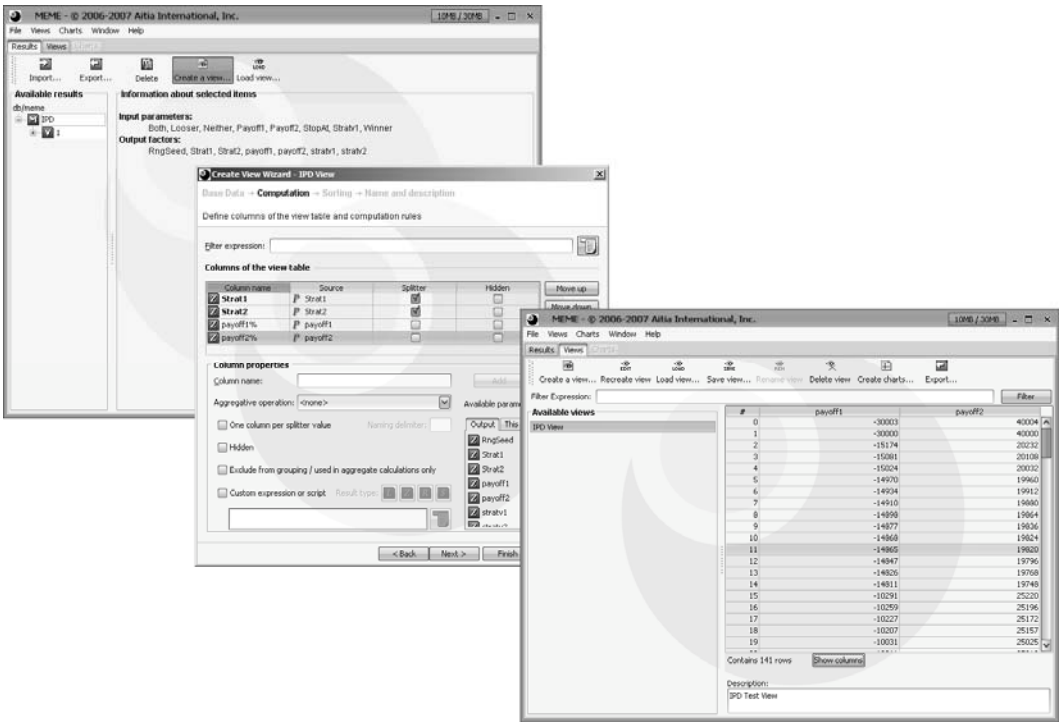

FIGURE 3 Creating a subset (views) of the original result data is done through a wizard. The modeler is presented with various filtering and computational options.

MEME provides the modeler with advanced functionality in creating subsets of the results database though. Conditional filtering, splitting (a generalized form of cross tabulation), aggregating and reorganizing data are supported, as well as the creation of derived statistics and custom computations typical of agent-based simulation, without any or with minimal coding. View creation settings can be saved as xml files for later use.

Once the data is organized into the desired form it can be exported for analysis in advanced statistical software of the user's choice, or it can be visualized through MEME's built-in Charting Wizard. The Charting Wizard is built on the Visualizations Package that is an ABM specific visualization tool developed by AITIA. It enables modelers to create visualizations for their simulations quickly and conveniently without any coding. The program offers visualizations[3] such as diagrams, grids and charts with both a vivid and rich design for presentation and a basic, black and white mode that suites scientific publication better.

Future Developments

MEME is not intended to be a sophisticated statistical tool. Although MEME already enables the user to export simulation results at any stage thus provide input as CSV files, further plans of development include implementing interoperation with such software.

Currently MEME is only capable to run parameter space explorations on a standard hyper-cube, thus allowing for interactive and incremental experiment design and execution. That is, a first set of runs is processed and their results are evaluated, which allows the user to determine the next set of runs. Ongoing developments include the introduction of intelligent parameter space exploration, where the explored sample of parameter space is irregular and is changing dynamically and adaptively.

The intellisweep capability is based on the meta-language we are currently working on, that describes what parameter combinations to explore next. The key to the workings of the intelligent parameter space exploration is the evaluation of results. Since this is always application dependent, general solutions are very limited. Therefore the advanced user is going to be provided with an integrated editor for developing call-back functions performing such evaluations in MEME.

Also general exploration routines are planned to be developed. The idea here is that the user defines a 'statement' about her results (i.e. the measured value converges/is always close to X, or depends linearly on Y, etc.). This is then converted into an evaluator that scores the measured value according its correspondence to the statement. The general exploration routines of MEME will then minimize this correspondence (maximize the error in the statement) using various and an extendable list of techniques (i.e. genetic algorithms, artificial neural networks, ant colony optimization, etc.) .

---

[3] List of currently available visualizations: Various 2D grids, bar charts, histograms, network and sub-graph visualizations, area-based chards, scatter plots, sequence visualizations and time series.

The Model Exploration Module is a modular system; all major components are decomposable and interchangeable. Hence in the future we would first like to extend MEME's import capabilities to reading result files from the MASON and JAS systems, then should demand arise we will also introduce parameter sweeps for the above mentioned, or any other Java-based modeling systems.

Acknowledgements

The work reported in this paper benefited from the partial support of the following grants:

- The GVOP-3.2.2-2004.07-005/3.0 (ELTE Informatics Cooperative Research and Education Center) by the Hungarian Government;
- The STREP project QosCosGrid (contract number 033883) by the European Commission;
- The AN FP6 Project Emergence in the Loop (EMIL) by the Information Society Technologies.

References

Axelrod, R. and Tesfatsion, L., 2006, "A Guide for Newcomers to Agent-Based Modeling in the Social Sciences", Appendix A in *Leigh Tesfatsion and Kenneth L. Judd (Eds.), Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics, Handbooks in Economics Series*, Elsevier/North-Holland, Amsterdam, the Netherlands.

Bankes, S. C. and Lempert, R., 2004, "Robust Reasoning With Agent-Based Modeling", *Nonlinear Dynamics, Psychology, and Life Sciences*, Vol. 8., Iss. 2, April, 2004, pp. 259-278.

Caromel, D., Delbé, C., di Costanzo, A. Ad Leyton, M., 2006, ProActive: an Integrated platform for programming and running applications on Grids and P"P systems, in *Computational Methods in Science and Technology*, Vol. 12., Iss 1, Polish Academy of Sciences, Poznan, Poland.

Gulyás, L., 2002, „On the Transition to Agent-Based Modeling: Implementation Strategies From Variables To Agents", *Social Science Computer Review*, Vol. 20, No. 4, Winter 2002, pp. 389-399.

Gulyás, L. and Bartha S., 2003, "How Much Is Too Much? -- What Programming Skills Are Really Needed to do ABM?" In *Proceedings of Seventh Annual Swarm Users/Researchers Conference*, Notre Dame, Indiana: Swarmfest 2003.

Gulyás, L. and Bartha S., 2005, “FABLES: A Functional Agent-Based Language for Simulations”, In *Proceedings of the Agent 2005 Conference on: Generative Social Processes, Models and Mechanisms*, Argonne National Laboratory, Chichago, IL, USA, October 2005.

Samuelson, D. and Macal, C., 2006, "Agent-based Simulation Comes of Age," *OR/MS Today*, Vol. 33, Number 4, pp. 34-38, Lionheart Publishing, Marietta, GA, USA.

Tatai, G., Gulyás, L., Laufer, L. and Iványi, M., 2005, “vBroker: Artificial Agents Helping to Stock Up on Knowledge”, In *Pechoucek, M., Petta, P. and Varga, L.Zs. (eds.). Multi-Agent Systems and Applications IV. Proceedings of the 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2005)*, pp. 336-345, Springer-Verlag.

Drone, http://drone.sourceforge.net/

NetLogo, http://ccl.northwestern.edu/netlogo/

Repast, http://repast.sourceforge.net/

Platform LSF, http://www.platform.com/Products/Platform.LSF.Family/Platform.LSF/

Swarm, http://www.swarm.org/wiki/Main_Page

SweepOver, http://www2.maths.ox.ac.uk/~challet/sweepover/sweepover.html