PROCEEDINGS OF SCIENCE

From Internal Validation to Sensitivity Test: How Grid Computing Facilitates the Construction of an Agent-Based Simulation in Social Sciences

Frank C.S. Liu[1]
Institute of Political Science, National Sun Yet-San University
No. 70 Lian-Hai Rd., Kaohsiung 804, Taiwan, R.O.C.
E-mail:csliu@mail.nsysu.edu.tw

Jing-Ya You
Academia Sinica Grid Computing Center
128, Sec.2, Academia Road, Nankang, Taipei 11529, Taiwan, R.O.C.
E-mail: jingya.you@twgrid.org

Yu-Ting Chen
Academia Sinica Grid Computing Center
128, Sec.2, Academia Road, Nankang, Taipei 11529, Taiwan, R.O.C.
E-mail: yuting.chen@twgrid.org

Jing-Lung Sun
Department of Computer Science, National Tsing Hua University
30013 No. 101, Section 2, Kuang-Fu Road, Hsinchu, Taiwan 30013, R.O.C.
E-mail: s9962571@m99.nthu.edu.tw

Over the past decades, we see a trend that social scientists adopt the experiment approach to study our social and political world. Particularly, agent-based modelling (ABM) is employed as a tool for “thought experiment” because theorists usually (1) fall short of empirical data to contrast with experiment results and (2) are more interested in solving theoretical puzzles than empirical puzzles. Applications of ABM in the social sciences, therefore, are commonly centered on issues that do not require a serious empirical validation process, at least compared to empirical data. Consequently, current application of ABM in social sciences (except the field of business management) has not reached the stage of sound validation and verification (V&V). To take a further step out of this situation, we suggest that researchers adopting computer-based approach conduct sensitivity test of their models. This step at least ensures that simulation results are robust and trust worthy. Few scholars in this field, however, are aware of the utility of grid and cloud computing that can advance the application of ABM. This paper is devoted to bridge the two sides (grid computing and political science) by, first, describing how grid and cloud computing may help the design of an agent-based model. Second, we will present an example of (re)constructing the SRAS model of political

[1] Speaker

PROCEEDINGS OF SCIENCE

preference formation—an agent-based model based on John Zaller's (1992) Receive-Accept-Sample theory, and put it into a sensitivity test with the power of grid computing.

Keywords: Agent-based modeling; empirical validation; empirical validation.

The International Symposium on Grids and Clouds and the Open Grid Forum
Academia Sinica, Taipei, Taiwan
March 19 - 25, 2011

Appendix I: Agent-Based Modeling Toolkits Evaluation Report

1. Comparison of ABM Toolkits

In the last decade, the development of ABM toolkits has grown rapidly. Important functionalities of ABM toolkits can be considered as below.

- Graphical visualization
- The ability to detach and re-attach graphical interfaces and to stop a simulation
- Moving models between computers for long simulations
- Statistical inference
- External GIS support for simulation with geography location dynamic

Below we report our evaluation of the most well-known ABM toolkits - Repast Simphony, Mason, Swarm, and NetLogo by factors that people concern most while developing their simulations. Note that Repast has few extensions, such as Repast Java, Repast Simphony, and Repast HPC. Repast HPC is the latest update which supports simulation on cluster or grid using openMPI, so that we think it's worthy to include in our study for such advance. We have compared these toolkits in the following matrix from programming to technical details of model construction perspective of view.

Table 1 presents basic information of the four ABM toolkits. The features that related to programming are listed in Table 2. The extendibility of these ABM toolkits with statistic tools and GIS, etc is listed in Table 3. Table 4 gives more information about how commonly used concepts and constructs in ABM differ across these toolkits.

Table 1 - Basic Information [4]

	Repast HPC	Repast Simphony	Mason	Swarm	NetLogo
Current Version	1.0 Beta	2.0 Beta	Version 14	2.2	4.12
License	New BSD	New BSD	LGPL	GPL	
Documentation	Limited (Manuals/ API)	Moderate (Manuals/API/MailList)	Moderate (Manuals/ API)	Moderate (Manual/API /Wiki)	(MailList/ Wiki)
User Base	Small	Largest	Increasing	Diminishing	Large
Modeling Language	C++	Java, Python	Java	Objective-C	NetLogo

Table 2 - Programming [4]

	RepastHPC	Repast Simphony	Mason	Swarm	NetLogo
Speed of Execution	Currently No GUI supported	Startup of GUI Simulation takes longer time than Mason; speed of simulation execution is in the same level of Mason	Fastest startup of GUI simulation		
Support for Graphical User Interface Development	N/A	Lack flexibility to customize	More flexible to modify and adjust	Displays of Line graphs, Histograms, Raster of 2-dimensional data	
Built-in Ability to Create Movies and Animations	N/A	2D/3D	2D/3D and able to export movie (Java JMF to create Movie)	2D	2D/3D
Ease of Learning and Programming	C++ experience	Java, Python and OO programmin g experiences required	Java and OO programmin g experiences required	Object-C experiences required	1. Good to use as a teaching tool 2. Highly recommend to people with little programmin g experience
Ease of Installation	Poor (Compilatio n required)	Easy (Eclipse Plugin)	Easy (Java Library)	Poor (Compilatio n required)	Very Easy

Table 3 – External Integration [4]

	RepastHPC	Repast Simphony	Mason	Swarm	NetLogo
Link to GIS	Not supportedyet	Limited (Support GIS projection, but with limited flexibility to customize in our experiment)	GeoMason: Geospatial Support for MASON	No extension for GIS	Support GIS extension; able to import vector GIS data
External Tools Support - R	Not supported yet	repast.simphony.R	No extension for R	No extension for R	No extension for R
External Tools Support - MATLAB	Not support yet	repast.simphony.matlab	No extension for MATLAB	No extension for MATLAB	No extension for MATLAB
External Tools Support - Others		*Agent Adaption using Neural Network (Java Object Oriented Neural Engine (Joone) (http://www.jooneworld.com))	extension for JUNG (popular social network system)		
Integration with Grid	**openMPI**	**Limited (Still not able to)**	**N/A**	**N/A**	**N/A**

Table 4 - Terminology/Terms [1]

Concepts	RepastHPC	Repast Simphony	Mason	Swarm	NetLogo
Model Construction	ShareConte xt and projection	Context and projection	Model		
Object that builds and controls simulation objects	model	model	model	modelswar m	observer
Object that builds and controls screen graphics	None	None	model withUI	observer swarm	interface
Object that represents space and agent locations		space/proje ction	field	space	world
Graphical display of spatial information	Shared spaces (network, grid and space)	projection (network, grid and space)	portrayal	display	View
User-opened display of an agent's state		Probe	inspector	probe display	monitor
An agent behaviour or event to be executed		Action	steppable	action	procedure
Queue of events executed repeatedly	schedule	schedule	schedule	schedule	forever procedure
Output dataset	Aggregated Data Collection (To aggregate output data across processes and write result in a single file in MPI code)				

2.More about Repast and Mason

Based on the glimpse of comparison, we picked up Repast Simphony and Mason as the best candidates for our current project of conducting sensitivity test. To make our final decision, we explored their functionalities and tried out their examples of Cellular Automata. We report our observation below.

2.1 Repast Simphony

Repast Simphony has the greatest functionalities of any ABM toolkits. It has supported wide range of external tools integration for statistical, neural network, and GIS, etc. Besides, it also provides numerical libraries, e.g. for random numbers. Models can be checkpointed in various formats including XML. However, as trying GIS tool in Repast Simphony, we found its limitation to extend the full functionality of GIS visualization and the difficulties on debugging with Repast Simphony. Thus, we consider visualizing the simulation results by using an external tool independently. The improvements in Repast Simphony compared to RepastJ are: a new GUI for developing models, an improved runtime GUI, and the addition of contexts and projections.

Repast Simphony provides contexts which can be arranged hierarchically and contains sub-contexts. Agents in a sub-context also exist in the parent context, but the reverse is not necessarily true. Projections can give the agents a space and can define their relationships. Projections are created for specific contexts and will automatically contain every agent within the context. Different projects can be made for different properties.

2.2 Mason

Mason is a lighter package compared to Repast Simphony. Mason has a clear focus on computationally demanding models with many agents executed over much iteration. Mason's developers aim to support general rather than domain specific tools. It was designed to serve as the basis for a wide range of multi-agent simulation tasks ranging from swarm robotics to machine learning to social complexity environments.

Regarding graphic visualization, MASON contains both a model library and an optional suite of visualization tools in 2D and 3D. The core models will run independently of visualization which can be added. The design of Mason has been driven largely by the objectives of maximizing execution speed and to assure complete re-producibility across hardware. In our tests, Mason has the fastest startup of a simulation. Checkpointing and migration of models is provided. The other advantage that MASON has also relates to batch runs, in that a simulation can be stopped on one machine, then copied to and restarted on another machine.

3. Conclusion

Mason has its strength and impressive performance on the visualization and animation. The flexibility of integration of GUI in Mason can be effective to meet different requirements of graphical displays and visualization for simulations. On the other hand, Repast Simphony has significant progress of development and also is the most comprehensive toolkits among all. The choice of tools will be regarding to complexity of models, displays and ease of implementation. Mason has some advantages on its flexibility and extensibility, especially of graphical animation and display. Overall, Repast Simphony has included very sufficient concepts to develop simulations. Although the comparatively complicated architecture of Repast Simphony might cause higher learning curve at the beginning, the effort will be worthy while a comprehensive simulation can be built. The extensibility and flexibility of UI display can be worked around by running batch mode and using simulation results, usually row data, as input for other statistics or analysis tools.