Environment Mediated Multi Agent Simulation Tools .

Conference Paper · October 2008
DOI: 10.1109/SASOW.2008.44 · Source: DBLP

3 authors, including:

Rym Zalila-Wenkstern
University of Texas at Dallas
65 PUBLICATIONS **872** CITATIONS
SEE PROFILE

Renee Steiner
Independent Researcher
9 PUBLICATIONS **59** CITATIONS
SEE PROFILE

Environment mediated Multi Agent Simulation Tools –A Comparison

S. Arunachalam, R. Zalila-Wenkstern, R. Steiner
*The University of Texas at Dallas, MAVs Lab**
{sxa066100, rmili, rsteiner}@utdallas.edu

Abstract

In this paper, we assess five tools that allow the specification and execution of Multi-agent based simulations. These tools are NetLogo, MASON, Ascape, RePastS and DIVAs

1. Introduction

Over the last decade, a plethora of MAS tools were developed and used for various purposes. In this paper, we focus on "framework and library" platforms that allow the specification and execution of Multi-agent based simulations. Tools in this category can be classified in two groups: those that emphasize the agent component and downplay the environment [20-22], and those that consider the environment an important component of the system and decouple it from the agents.

In this paper, we discuss five tools that acknowledge the importance of the environment in a multi-agent-based simulation system. These are *NetLogo* [2], *MASON* [3–5], *Ascape* [6-9], *RePastS* [10-16], and *DIVAs* [17]. Our choice is based on the fact that according to the literature, NetLogo, MASON, Ascape and RePastS are considered to be among the most effective platforms in the market [1]. Since much attention has been given to agents, we focus on assessing the tools from the perspective of the environment, and the interactions between the agents and the environment. To provide a basis for comparison, in each tool we have attempted to specify and execute a social simulation model where agents represent "humans", and the environment the world.

In the next section, we give a brief overview of the various platforms. In Section 3, we discuss the criteria used to compare them. This is followed by a description of the social simulation case study, and an evaluation of the tools.

2. The Tools

NetLogo [2] is a programmable platform for simulating models related to natural and social phenomena. MASON (Multi-Agent Simulation of Networks) [3-5], has been designed to provide simulations for applications such as swarm robotics, machine learning, social complexity. Ascape [6-9] is a framework designed to support the development, visualization, and exploration of agent based models. Ascape is designed mainly for social science simulations. RePast (Recursive Porous Agent Simulation) [11] was initially developed to support social science applications. The latest version is RePast Simphony (RePastS) [10] and can be used to simulate a variety of applications (e.g., network simulations, GIS applications). DIVAs (Dynamic Information Visualization of Agent systems) [17] is a platform developed by our research group at the University of Texas at Dallas. It includes a specification and a simulation tool that run in the Eclipse IDE. So far, DIVAs has been predominantly used for social simulations.

3. Comparison Criteria

The assessment criteria used to evaluate these tools are divided into four groups: *design criteria*, *model specification criteria*, *model execution criteria*, and *documentation*. For the sake of conciseness, not all factors are described in this paper. A detailed description of the factors can be found in [24]. Each criterion is evaluated using either a discrete four-rating scale, or a quantitative metric, when possible.

3.1 Design Criteria

These criteria describe the design decisions used in the development of the various types of environments. We have identified three design evaluation criteria.

1. Environment structural complexity.

A designer may decide to model environments using various underlying structures. For example, an environment can be modeled as a graph, a grid, a continuous space or a combination of these. It is clear that a tool offering a simple grid-based environment is limited in terms of the applications it can model. More complex structures such as *continuous space* or *hybrid structures* (e.g., combination of graph and continuous space) can be used to represent more intricate realistic applications. Hence, the following ratings for structural complexity are shown below in Table 1.

Low	Medium	High	Very High
Grid	Graph	Continuous Space	Hybrid

Table 1. Rating for environment structure complexity

*Supported by Rockwell-Collins

2. Environment distribution

This factor takes into account both *structural distribution*, and *processing distribution*. An environment is structurally distributed if, at any point in time, no centralized entity has a complete knowledge of the state of the environment as a whole. An environment is distributed from a processing perspective if it is designed to be executed in a distributed network.

A tool that offers distributed environments from a structural and processing perspective will be assigned a high rating for this factor as presented in Table 2.

Low	Medium	High	Very High
No distribution	Distributed processing	Distributed structure	Distributed structure & processing

Table 2. Rating for environment distribution

3. Agent and Environment coupling

This factor evaluates the amount of environment information an agent has to carry. From a design perspective, coupling is an undesirable feature. Hence, as shown in Table 3, the more coupled the agent and environment are, the lower the rating

Low	Medium	High	Very High
Very high coupling	High coupling	Average coupling	Low coupling

Table 3. Rating for agent-environment coupling

3.2. Model Specification Criteria

The criteria described in this section are intended to assess the ease of use of the specification tool. This is achieved by evaluating a) the ease of specifying the environment as an independent component, and b) the amount of environment information that needs to be specified in an agent.

1. Ease of specifying the environment

This criterion is assessed using three factors: the amount of information that can be specified through the user interface (UI), the expected level of programming skills, and the effort spent creating the base environment in our case study.

Specification Features offered by UI

This criterion measures, how much can be specified using the UI. If the entire environment can be specified using graphical features such as drag and drop, wizards (e.g., for importing images), etc., then the tool is rated *very high*. If the user can specify some aspects of the environment through the UI and the rest need to be programmed, then the tool is rated *high*. A tool is rated *medium*, if its UI allows the specification of only a few simple models while the complex ones need to be programmed. A tool is rated *low*, if the user cannot specify anything about the environment through the UI. This criterion is summarized in Table 4.

Low	Medium	High	Very High
No feature can be specified	Simple environment. model features can be specified	Most features can be specified using the UI	All features can be specified using the UI

Table 4. Rating for quality of user interface

Level of programming skill and Effort required to create the base environment

A detailed description of these factors can be found in [24].

2. Specified Environment knowledge in Agents

The tools that expect users to specify environment information in agents are rated low as presented in Table 5. This criterion is related to the agent-environment coupling criterion discussed in Section 3.1. If the tool has been designed with high coupling between the agents and the environment, more information needs to be specified about the environment in the agents.

Low	Medium	High	Very High
Both the node and coordinate information are provided to the agents	The existence of the node in the environment is specified and the location of the node is read at run time	The agent only has the knowledge of the node existence and does not know about their location in the environment	Not pre-specified in agents

Table 5. Rating for environment knowledge in agents

3.3. Model Execution Criteria

The criteria used to assess the execution of a simulation are: a) the quality of the visualization, b) the simulation views, and c) how easy it is to change properties of the model at execution time.

1. Quality of the visualization

As shown in Table 6, this factor is given the rating *very high* if the tool provides excellent image rendering of the agents and the environment.

Low	Medium	High	Very High
Poor image rendering of agent movement and environment	Poor image rendering of agent movement, Good for environment	Good image rendering of agent movement and environment	Excellent image rendering of agent movement and environment

Table 6. Rating for visualization quality

2. Simulation view

Different views of the simulation are essential to study the agent behavior and its effect on the environment. It is also essential if a user needs to inspect a particular agent or a particular area in the environment. A very high rating is given for tools that offer 2D and 3D views and also allows for inspection of the agent and environment. Table 7 details the rating scale for simulation views.

Low	Medium	High	Very High
2D, non-toroidal	Visual 2D, programming for 3D, programming for toroidal space	2D, inspection, programming for 3D,Toroidal space	Provide 2D, 3D, rotations, inspection, etc. Choice between toroidal and non-toroidal environment space

Table 7. Rating for simulation view

2. Ease of model property modification

The ratings for this criterion are given in Table 8. A tool that offers the capability to change parameters of the model at execution time without stopping the simulation (i.e., interactive simulation), and observe the effect of such a change is given a very high rating.

Low	Medium	High	Very High
Cannot modify properties during simulation. Need to program the changes.	Cannot modify properties during simulation. Make changes and re-run	Can modify properties without stopping simulation. Have to re-run simulation to see changes	Can modify properties without stopping simulation. Changes take effect immediately.

Table 8. Rating for change of properties of a model

3.4. Documentation

The documentation is assessed in terms of the *quality* and *effectiveness* of the documentation and tutorials.

1. Quality of the documentation and tutorials

A detailed description of this factor can be found in [24].

2. Effectiveness of the documentation and tutorials

The time spent installing and understanding the tool is a good indicator of the documentation and tutorial effectiveness. Hence, this factor is evaluated in time metrics.

4. Case Study

Our goal is to specify and execute a simple simulation model in which the environment is represented as a graph, and agents as entities moving along the graph in the course of achieving their goals. A concrete example of this model is a social simulation application where the environment represents the world, and agents represent "humans" (see Figure 1). In this case, nodes correspond to physical locations, and edges are pathways between nodes. We assume that the environment is dynamic in the sense that during a simulation, changes can be made to the graph. The purpose of the simulation is to study the reaction of agents to environmental factors.

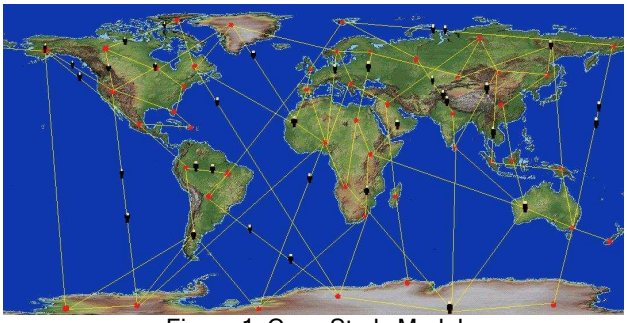

Figure 1. Case Study Model

In order to specify the environment, the user has to upload the world map, and specify nodes and edges. During the execution of the simulation, the user may add and/or remove nodes and edges. These changes should take effect immediately, and the new environment state should be passed on to the appropriate agents.

5. Tools Assessment

In this section, we evaluate *NetLogo, MASON, Ascape, RePastS, and DIVAs* with respect to the model discussed in Section 4, and using the criteria discussed in Section 3.

5.1 Design Criteria

1. Environment structural complexity

NetLogo provides the user with only a grid structure of the environment. An individual cell in the grid is called a patch [2]. Patches can be sensed or not sensed by the agents, based on the user's choice of the model. Hence, NetLogo is given a rating of *low*. MASON is rated *high* with respect to the environment structure since it provides the user with grid, continuous space and network environment structures [4]. The graph is called a *network* in MASON. The environment structure in Ascape has been rated *medium*. Ascape offers grid and graph environment structures. An individual lattice is called a *cell* and the agents that interact with the cell are called *cell occupants* [7]. In the graph structure, the cells are the nodes and the cell occupants move from node to node. RePastS environment structures are rated *very high*. RePastS offers continuous space, grid, network, geography, and scalar field structures [14]. The

geography environment structure can be extremely useful for GIS based models. DIVAs has been rated *medium* as it offers a graph based environment structure [17], which has nodes and edges. Agents move along these edges.

2. Environment distribution

NetLogo has been rated *very high* for its environment distribution capabilities. NetLogo provides users with HubNet [2] which enables the same model simulation to be controlled by multiple users and, hence, the entire simulation is distributed with respect to architecture and processing. MASON and Ascape have been rated *low* as they offer no distribution capabilities in terms of the processing and structure. RePastS has been rated *medium* as it possesses only distributed processing capabilities. The environment structure is not distributed. DIVAs has been rated *high* as it has a distributed environment structure but has limited distributed processing capabilities.

3. Agent and Environment coupling

NetLogo has been rated *high* as the agent is aware only of the existence of a patch and its activities on the environment. The agent does not possess any information about the resources available in the patch. MASON has been rated *low* as the agent carries information of the entire environment. For an agent to walk along an edge or even to be placed in the graph, the coordinates of the node position must be programmed into the agent. Ascape has been rated *low* in terms of coupling since the user must specify the node positions within the agents. RePastS has been rated *low to medium* for this criterion as the environment and agent can be decoupled for simpler models. However, for our case study, the agents included environmental information to enable their movement about the graph. This makes the model tightly coupled. DIVAs has been rated *very high* as the agents are completely decoupled from the environment.

5.2. Simulation Specification Criteria:

1. Ease of specifying environment

Specification Features offered by UI

NetLogo has been rated *high.* The UI contains procedures for the specification and simulation of the model in different tabs. The environment image can be uploaded by choosing the "import world" option [2], but the user cannot specify the nodes and edges graphically. In MASON and Ascape the user must program the environment model. Hence, the rating is *low*. RePastS has been rated *high* in this category since the user partially specifies the environment by filling in properties in the model file. DIVAs has been rated *very high* in this category as the user only has to enter information in tables in the specification tool.

Level of programming skill:

NetLogo has been rated *high* in terms of the programming skills required by the user. It has a simple programming language that is very easy to understand, and easy to use to build models. MASON has been rated *low* as it requires the user to program everything in the model including the visualization. Intensive programming is involved if the user chooses to create edges between specific nodes. Ascape is rated *medium*, as it requires some basic programming skill from the user. In order to specify the environment and the agents, the user has to program. RePastS is rated *low* for this criterion as the user has to program the environment. Importing the image of the map is also a challenge in RePastS. DIVAs is rated *very high* since there is no programming involved in specifying the model. The entire model is specified using a graphical editor.

Effort required to create the base environment.

A detailed description of this factor can be found in [24].

2. Specified Environment knowledge in Agents

NetLogo has been rated *high* in this category as the agents are aware of only the existence of the patches and not the location of the patches. MASON has a rating *low* as the agent must hold the information of the node existence as well as the position of the node in order to move in a model. Ascape also has a rating of *low* as the user must specify the node position in which the agents must exist and also move. RePastS has been rated *very high* as the agents only include information that allows them to detect whether they are on a node or an edge. DIVAs has been rated *very high* as the no information about the environment is specified in the agents.

5.3 Simulation Execution Criteria:

1. Quality of the visualization

NetLogo has been rated in between *medium* and *high*. The image in the simulation is updated continuously for the environment, but the agent jumps from node to node rather than moving along the edges. MASON has been rated *medium* for its image rendering capability in the environment. The agent movement is similar to that of NetLogo. The user can program for agent movement along the edges by identifying their inclination with respect to the x and y axis of the environment. Ascape has been rated *medium* as the capabilities are similar to MASON. RePastS has been rated *high* in this category as the agent movement in a Geography structure from one node to another is along the edges and the environment is

also rendered continuously. *DIVAs* is also rated *high* as the agent movement and the image rendering capabilities are similar to RePastS.

2. Ease of change of properties of model

NetLogo has been rated between *medium* and *high* with respect to the modifiability of the model. The behavioral changes to the model can be made by typing in commands at the Observer pane [2]. But changes to the model such as removing a node from the model would involve deleting the information from the program and reinitializing the model. MASON has been rated *low* as the user must modify the program and compile it before executing it again for the changes to take effect. Ascape also has been rated *low* since no modifications can be made without reprogramming the model and starting the simulation again. This is a major handicap of the tools that require programming. RePastS has been rated *high* as it allows the user to modify the model while the simulation is running, the model must be started again for the changes to take effect. DIVAs has been rated *very high* in this category. The user can make changes to the model, add and remove agents and nodes without the need to stop the simulation.

3. Simulation view

NetLogo has been rated between *high* and *very high* for this criterion as it offers 2D and 3D views of the model, but the nature of the environment space is toroidal by default. MASON has been rated *medium* as it offers a basic 2D visualization. If 3D view is desired, the user must extend Java 3D and program for the 3D view. The grid structure is toroidal, the continuous space and network structure are non-toroidal, but can be programmed to be toroidal [4]. Ascape offers only 2D visualization and no 3D visualization. The environment space is toroidal by default, but can be programmed to be non-toroidal and, hence, has been rated between *low* and *medium* [6]. RePastS has been rated *very high* as it offers 2D, 3D visualizations and allows the user to choose the nature of the space in graph and grid structures. The Geography structure is non toroidal. DIVAs has been rated as *low* for this criteria as it offers only a 2D visualization and the environment space is non-toroidal.

5.4. Documentation

1. Quality of the documentation and tutorials

Table 14 summarizes the evaluation of this criterion. A detailed discussion can be found in [24].

2. Effectiveness of the documentation and tutorials

The effort spent in understanding NetLogo was 10 Person Days out of which 5 Person Days involved understanding how the tool works and by learning the commands. The other 5 Person Days were spent on working with the tutorials. The effort spent in understanding MASON was 10 Person Days, out of which 4 Person Days were spent on understanding the tool and 6 Person Days were spent on the tutorials. The effort spent in Ascape was 12 Person Days, out of which 5 Person Days were spent in understanding the tools. Since the tutorials were minimal, 7 Person Days were spent in studying all the sample models provided with the tool. The effort spent on RePastS was 12 Person Days. 7 Person Days were spent in understanding the tool and 5 Person Days were spent in working with the tutorials. The effort spent on DIVAS was 10 Person Days. All Person Days were spent on understanding the tool, which was due to the lack of proper tutorials.

6. Conclusion

Our results are summarized in Tables 9-12.

	Structure Complexity	Distribution	Coupling
NetLogo	Low	Very high	High
Mason	High	Low	Low
Ascape	Medium	Low	Low
RePastS	Very high	Medium	Low – Medium
DIVAs	Medium	High	Very high

Table 9. Summary of Assessment using Design Criteria

	User Interface	Programming skill	Agent's knowledge of environment
NetLogo	High	High	High
Mason	Low	Low	Low
Ascape	Low	Medium	Low
RePastS	High	Low	Very high
DIVAs	Very high	Very high	Very high

Table 10. Summary of Assessment using Simulation Specification Criteria

	Visualization quality	Modifying the model	Simulation view
NetLogo	Medium – High	Medium – high	High –very high
Mason	Medium	Low	Medium
Ascape	Medium	Low	Low - medium
RePastS	High	High	Very high
DIVAs	High	Very high	Low

Table 11. Summary of Assessment using Simulation Execution Criteria

	Quality of documentation and tutorials
NetLogo	Very high
Mason	Very high
Ascape	Medium – high
RePastS	High
DIVAs	Medium

Table 12. Summary of Assessment using Quality of Documentation Criteria

Based on these results, we conclude that if the user does not have a programming background, the most preferred choices are clearly DIVAs, followed by NetLogo, Ascape, RePastS and MASON respectively. The next most important criterion for choosing a tool would be the complexity of the environment. The more choices for environment structures, the broader its application in the field of multi-agent simulation systems. RePastS would be the most preferred choice based on the environment structure, followed by MASON and then the rest of the tools. Based on the distribution criteria of the tool, the most preferred tool would be NetLogo, followed by DIVAs and RePastS and the other two tools. Taking into account all criteria discussed, the tool ranking would be NetLogo, DIVAs, RePastS, MASON and Ascape respectively.

7. Acknowledgements

We are thankful to Seth Tisue (NetLogo), Sean Luke (MASON), Miles Parker (Ascape), Michael North (RePastS) and others for their guidance in working with the tools.

8. References

[1] S.F Railsback, S.L Lytinen, and S.K.Jackson, "Agent-based Simulation Platforms: Review and Development Recommendations", *Simulation*, vol. 82, 2006

[2] U. Wilensky, NetLogo, http://ccl.northwestern.edu/netlogo/ Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

[3] MASON, http://cs.gmu.edu/_eclab/projects/mason/ George Mason University

[4] MASON documentation, http://cs.gmu.edu/~eclab/projects/mason/docs/#docs

[5] S Luke, C Cioffi-Revilla, L Panait, and K Sullivan, "MASON A New Multi-Agent Simulation Toolkit, Department of Computer Science and Center for Social Complexity", *In Proceedings of SwarmFest*, Michigan, USA, 2004

[6] Ascape, http://ascape.sourceforge.net/index.html

[7] Ascape documentation, http://ascape.sourceforge.net/index.html/#Documentation

[8] M.T.Parker, "What is Ascape and Why Should You Care": *Journal of Artificial Societies and Social Simulation*, vol. 4, no. 1, January 2001

[9] M E. Inchiosa and M T. Parker, "Overcoming design and development challenges in agent-based modeling using ASCAPE." *Proceedings of National Academy of Sciences (PNAS) of United States of America*, vol 99, May 2002.

[10] RePastS, http://repast.sourceforge.net/ (Argonne National Laboratory Decision and Information Sciences Division Center for Complex Adaptive Agent Systems Simulation).

[11] RePast3, http://repast.sourceforge.net/repast_3/index.html

[12] M.J North., E. Tatara, N.T. Collier, and J. Ozik, "Visual Agent-based Model Development with Repast Simphony", *In Proceedings of the Agent 2007 Conference on Complex Interaction and Social Emergence, Argonne National Laboratory*, Argonne, IL USA, November 2007.

[13] E. Tatara, M.J. North, T.R. Howe, N.T. Collier, and J.R. Vos, "An Introduction to Repast Modeling by Using a Simple Predator-Prey Example", *In Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects, Argonne National Laboratory,* Argonne, IL USA, September 2006.

[14] T.R Howe, N.T. Collier, M.J. North, M.T. Parker, and J.R. Vos, "Containing Agents: Contexts, Projections, and Agents", *In Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects, Argonne National Laboratory,* Argonne, IL USA, September 2006

[15] J.Ozik, M.J. North, D.L. Sallach, and J.W. Panici, "ROAD Map: Transforming and Extending Repast with Groovy," *In Proceedings of the Agent 2007 Conference on Complex Interaction and Social Emergence, Argonne National Laboratory*, Argonne, IL USA, November 2007.

[16] M.J North, T.R. Howe, N.T. Collier, and J.R. Vos, "Repast Simphony Runtime System," in C.M. Macal, M.J. North, and D. Sallach (eds.), *In Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms, ANL/DIS-06-1*, October 2005.

[17] R.Z. Mili, R Steiner, E Oladimeji, "DIVAs: Illustrating an Abstract Architecture for Agent-Environment Simulation Systems", *Multi agent and Grid Systems, Special Issue on Agent-oriented Software Development Methodologies* 2 (4), 2006.

[18] DIVAs, http://mavs.utdallas.edu/index.php

[19] R. Steiner, DIVAS Developer's Manual, March 2008.

[20] DECAF, http://www.cis.udel.edu/~decaf/

[21] John Graham, Victoria Windley, Daniel McHugh, Foster McGeary, David Cleaver, and Keith Decker, "Tools for Developing and Monitoring Agents in Distributed Multi Agent Systems", *Workshop on Agents in Industry at the Fourth International Conference on Autonomous Agents*, Barcelona, Spain, June, 2000

[22] JACK Documentation, http://www.agent-software.com/products/jack/documentation_and_instructi/jack_documentation.html

[23] Nick Malleson, RePast City Tutorial, http://portal.ncess.ac.uk/access/wiki/site/mass/repastcity.html University of Leeds

[24] S. Arunachalam, Environment Mediated Multi Agent Tools–A Comparison, *Masters Thesis*, University of Texas at Dallas, Fall 2008.

[25] Renee Steiner, Engineering Open Environments for Multi – Agent Simulation Systems, *Ph.D. Thesis*, University of Texas at Dallas, Fall 2006.