ISSN 2277-3061

Study and Comparison of Four Agent-Based Simulation Tools: Repast, SeSAm, Netlogo and GAMA

Wafa Mefteh
wafa.mefteh07@gmail.com

ABSTRACT

Several simulation tools have been proposed in the literature and many surveys have been realized on this. The objective of this paper is not to give a survey but to compare four principal tools which are Repast, SeSAm, NetLogo and Gama according to some evaluation criteria that I consider important when designing Adaptive Multi-Agent Systems.

Indexing terms/Keywords

Simulation Tools; Agent-Based Simulation Tools; Adaptive Multi-Agent Systems.

Academic Discipline And Sub-Disciplines

Computer sciences.

SUBJECT CLASSIFICATION

Modelling and simulation.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATION JOURNAL OF COMPUTERS AND TECHNOLOGY

Vol. 13, No. 4

editorijctonline@gmail.com

www.cirworld.org/journals

INTRODUCTION

There is some ambiguity concerning the terms "platform" and "toolkit (tool)" but generally they have been used interchangeably. In this paper, I also use these terms interchangeably. Several surveys have been done on agent simulation tools. The purpose of my work is not to present a state of the art about agent based simulation tools but to compare four principal tools according to some criteria defined later in this paper. The objective is to select the more appropriate tool to Adaptive Muli-Agent Systems (AMAS) [1] [2]. The proposed criteria are not determinist for any intended use of the simulation tool. These criteria was defined according to some characteritics that I find necessary to build simulations for Adaptive Multi-Agent Systems [3].

In this paper, I beguin by presenting the different criteria to be used. Then I describe each tool according to these criteria and Finally I give a synthesis about the comparison of these tools.

CRITERIA OF COMPARISON

In this section, I describe eight criteria of comparison (table 1). Some of these criteria are inspired from [4] and the others are proposed according to my objective from realising simulations. For each criterion, I define a set of values which can assigned to this criterion. For each value, I give a rate according to my objective from verifying the criterion. For example, if I consider a criterion **C** which can has as values: *c1* and *c2*. If *c1* interests me more than *c2*, I assign a more important rate to *c1* than *c2*. In the following, I use four rates which are:

rate	Used for
0	Non-interesting values. The tool which has this value, for one or more criteria, cannot be selected.
20	Very quite interesting values.
50	Quite interesting values.
100	interesting values.

Table 1 describes the different criteria of comparison which I will use them later to compare the four tools. For each criterion, I give the possible values that this criterion can obtain and the rate assigned to each value.

Criteron Name	Criteron Description	Possible Values	Values Rates
Availibility	Is the tool free and open source or not ? The availability of the code source enables to integrate new plug-ins if necessary to well simulate agents in some cases.	V1 = Free and available. V2 = Free and not available.	Rate of V1 = 100. Rate of V2 = 0.
Project Activity	Does the tool is used in research ? This criterion enables to evaluate the level of professionalism of the tool. Taking advantages from other users will probably help to better exploit it.	V1 = Used in research. V2 = Not used in research.	Rate of V1 = 100. Rate of V2 = 20.

Complexity Level	This criterion is to evaluate the level of complexity to handle the tool. This includes the language required to develop a model and to run a simulation, and if there are sufficient documentation and demonstration simulation models. By documentation, I am looking for manuals that explain how to use the toolkit. By demonstration simulation models, I am looking for examples of building and running simulation models. The existence of a well documentation and demonstration simulation models help to quickly understand the tool and how to build a simulation.	V1 = Simple.
		V2 = Medium.
		V3 = Complex.
			Rate of V1 = 100.
			Rate of V2 = 50.
			Rate of V1 = 0.

Generality	Most of tools are dedicated to a particular field of applications. I am interested by a tool that is as general as possible to solve different types of complex systems which	V1 = Specialized.
		V2 = Litle specialized.
		V3 = General.
			Rate of V1 = 0.
			Rate of V2 = 20.
			Rate of V1 = 100.

	can be related to different domains.		

Possibility of Using Diagrams	This criterion permits to know if I can design and orchestrate simulation experiments via diagrams. It is an interesting criterion because the possibility of using diagrams (UML like) to describe the system to be simulated will further facilitate the task of the designer using the simulation tool.	V1 = With diagrams.
		V2 = Without diagrams.
			Rate of V1 = 100.
			Rate of V2 = 20.

Dynamic Tuning	This criterion indicates if the simulation parameters can be modified in real time. It will be interesting for the designer if he can modify the simulation parameters when simulation is running in order to converge the system towards the functional adequacy.	V1 = Dynamic tuning.
		V2 = No Dynamic tuning.
			Rate of V1 = 100.
			Rate of V2 = 0.

Measurement	This criterion is to verify if the tool proposes or not analysis mechanisms. I am interested by a tool that has analysis tool integrated in order to help the designer to analyse the behaviours of the agents.	V1 = With analysis.
		V2 = Without analysis.
			Rate of V1 = 100.
			Rate of V2 = 0.

Agents Type	This criterion verifies if the tool enables the simulation of communicating agents or situated agents or both [5]. To simulate Adaptive Multi-Agent Systems, I am interested to Communicating agents.	V1 = Only situated agents.
		V2 = Only communicating agents.
		V3 = Communicating and situated agents.
			Rate of V1 = 0.
			Rate of V2 = 90.
			Rate of V3 = 100.

GENERAL DESCRIPTION OF REPAST, SESAM, NETLOGO AND GAMA

In this section, I give a general description of Repast, Sesam, Netlogo and Gama. For each tool, I describe the different elements of comparison proposed in the previous section.

Repast (http://repast.sourceforge.net/repast_3) was developed at the University of Chicago's Social Science Research Computing Lab specifically for creating agent-based simulations in social sciences. It borrows many concepts from the Swarm which was the first re-usable software tool created for agent based modelling and simulation. Swarm was developed at the Santa Fe Institute in 1994 (for more information, see http://www.swarm.org) .

Repast simulations typically have at least two classes:

- Agent class: describes the behaviour of the agents.
- Model class: coordinates the set-up and running of the model. s

Table 2 describes Repast according to the different defined elements of comparison.

Criteron Name	Description	Assigned Note
Availibility	Repast is free and open source.	100
Project Activity	Repast has been under continuous development for over 10 years and it was used by many researchers.	100
Complexity Level	Repast supports Java programming. Repast models can be developed in many languages including Java, C sharp, Managed C++, Visual Basic.Net, Managed Lisp, Managed Prolog, and Python scripting. Repast provides few documentation and few demonstration simulation models.	0
Generality	Repast was clearly intended to support principally one domain which is "social science" and includes tools specific to this domain.	20
Possibility of Using Diagrams	Repast enables the description of agent behaviours via a flowchart representing the different properties of the agent and the sequence of the different tasks realized by the agent during its life cycle.	100
Dynamic Tuning	Repast enables users to dynamically access and modify agent properties, agent behavioural equations, and model properties at run time.	100

Measurement	Repast does not integrate analysis mechanisms.	0
Agents Type	Repast enables the simulation of situated agents.	0

Sesam (http://www.simsesam.de/) is a generic environment for the development and simulation of Multi-Agent models. The main focus is to enable to construct models by visual programming. Sesam was developed at the University of Wurzburg and applied in several projects in different application domains.

The main entities in a Sesam model are:

- Agents: they consist of a body containing body variables and have a behaviour described by activity diagrams.
- Resources: they are like agents, except that they don't have a behaviour and they should be used for passive entities in the model.
- The World: it is a kind of agent class with a special role in interaction: it forms the basis for the representation of all environmental dynamics and structure that can not be captured in agents and resources.sss

Table 3 describes Sesam according to the different defined elements of comparison.

Criteron Name	**Description**	**Assigned Note**
Availibility	Sesam is free and open source.	100
Project Activity	Sesam is used by many researches and it is used in education.	100
Complexity Level	Sesam supports visual programming. It has graphical-based programming capabilities which are much more simple to learn and use than traditional programming languages. Sesam provides several documentations, tutorials ans demonstration simulation models and tutorials. It has also project wikis as part of user support.	100
Generality	Sesam is a general purpose agent based platform. It is not geared toward special domains. It is also oriented towards teaching computer simulation.	100
Possibility of Using Diagrams	Sesam provides the possibility of describing the agent behaviour using an activity diagram. Each activity is then defined using a visual programming. A list of configurable functions allows to link actions to nodes and conditions to transitions between nodes. The design of agents becomes very quick and intuitive.	100
Dynamic Tuning	Sesam permits to realize experimentations which means the possibility of launching many simulation simultaneously and modifying the value of one or more parameters. It is possible to modify the simulation parameters during runtime.	100
Measurement	Sesam has an analysis editor which allows to create an analyser that monitors the simulation. The simulation results can be displayed as graphs or stored in a file in tabular form.	100
Agents Type	Sesam enables the simulation of both situated and communicating agents.	100

NetLogo (http://ccl.northwestern.edu/netlogo/) was first created in 1999 by Uri Wilensky at the Center for Connected Learning and Computer-Based Modeling, then at Tufts University in the Boston area. Netlogo is a multi-agent programming language and modelling environment for simulating natural and social phenomena.

Netlogo manipulates:

- Turtles: which are agents.
- Patches: objects in turtles environment.

Table 4 describes Netlogo according to the different defined elements of comparison.

Criteron Name	Description	Assigned Note
Availibility	Netlogo is free and open source.	100
Project Activity	Netlogo is used by thousands of students, teachers and researchers worldwide.	100
Complexity Level	Netlogo supports programming in Logo Dialects extended to support agents. It is built-in graphical interfaces and comprehensive documentation. In spite of Repast, Netlogo has extensive documentation, tutorials and a models library which is a large collection of pre-written simulations that can be used and modified.	50
Generality	Netlogo was intended as an educational tool. It helps beginning users getting started authoring models. Netlogo has a primary specialization towards the social and natural sciences.	20
Possibility of Using Diagrams	Netlogo does not offer the possibility of describing the agents behaviours via diagrams.	20
Dynamic Tuning	Netlogo does not provide the possibility of changing simulation parameters at runtime. It should stop the simulation then adjust the "tolerance" slider to a new value then start the model running again.	0
Measurement	Netlogo does not integrate analysis mechanisms.	0
Agents Type	There really is no agent in NetLogo. This platform permits only to create a program for manipulating objects in a two-dimensional space.	0

Gama (https://code.google.com/p/gama-platform/wiki/GAMA?tm=6) is a simulation platform that integrates geographical information data. It aims at providing field experts, modellers, and computer scientists with a complete modelling and simulation development environment for building spacially explicit multi-agent simulations. It is being developed by several French and Vietnamese research teams under the umbrella of the IRD/UPMC International Research Unit UMMISCO since 2007.

The Gama model is composed of four sections:

- Global: the global agent of a Gama model.
- Entities: the definition of the different species which are used to specify the structure and behaviours of agents.
- Environment: contains definitions of environments. Gama supports three types of topologies for environments: continuous, grid and graph.
- Experiment: defines experiments to run.

Table 5 describes Gama according to the different defined elements of comparison.

Criteron Name	Description	Assigned Note
Availibility	Gama is free and open source.	100
Project Activity	Compared with the tools presented above, it is somewhat less used by other researchers. Gama provide several documentation and demonstration simulation models. Gama provides also video presentations and many tutorials.	100
Complexity Level	Gama provides a complete modeling language, GAML, for modeling agents and environments. It is a language simple to use. Gama provides several documentations, tutorials ans demonstration simulation models. It has also project wikis as part of user support.	50
Generality	Gama is a generic platform which permits the simulation of many different systems.	100
Possibility of Using Diagrams	Gama does not enables the description of the agent behaviour via diagrams. But, it permits to define in graphical mode a simulation as a whole with the environment, entities, etc.	20
Dynamic Tuning	Gama permits the modification of simulation parameters at runtime.	100

Measurement	Gama provides analysis mechanisms in batch mode.	100
Agents Type	Gama supports communicating agents.	100

SYNTHESIS

According to this comparison (figure 1), Sesam looks as an excellent tool to simulate Adaptive Multi-Agent Systems. It is a free and simple tool which enables an easily implementation of the agents. In Sesam, the agent behaviour is implemented as a set of activities and transitions between these activities and it provides a visual programming which facilitates the addition of new plug-ins to add new functionalities. Sesam enables the implementation of situated and communicating agents (other tools as Netlogo and Repast permit only the implementation of one category of agents) and it offers tools to analyse a simulation. An other important characteristic of Sesam is that it is possible to modify the data values of the agents, of the environment and of the resources during the simulation. It is a very important characteristic in our sens because it enables the designer to interact with the simulation in order to converge the behaviour of the system to the functional adequacy. Sesam provides a tool for the easy construction of complex models, which include dynamic inter-dependencies or emergent behaviour.

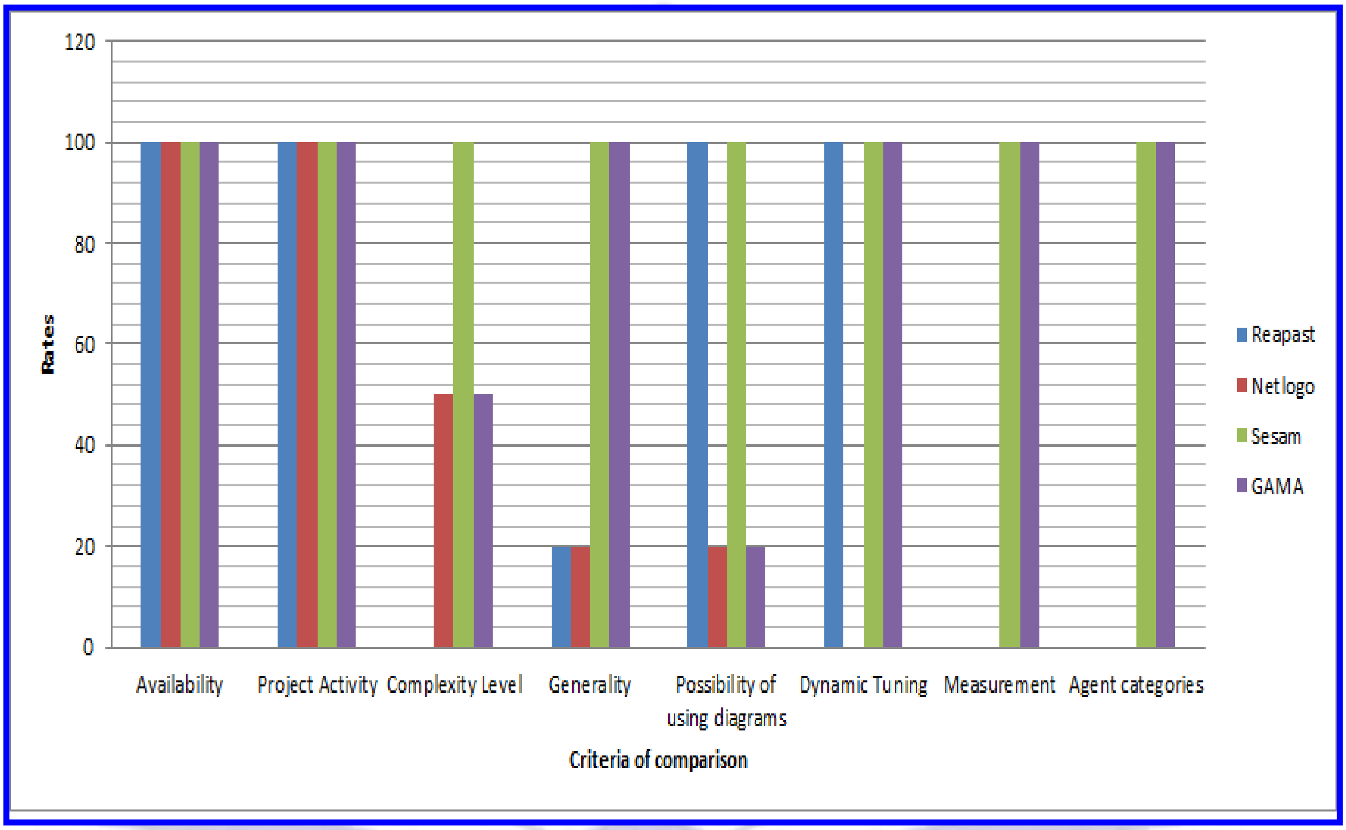

Figure 1 : Results of the Comparison.

REFERENCES

[1] Georgé, J-P., Gleizes, M-P. and Glize, P. 2003. Conception de systèmes adaptatifs à fonctionnalité émergente: la théorie des AMAS. In “Revue d’Intelligence Artificielle”.

[2] Glize, P. 2001. L’Adaptation des Systèmes à Fonctionnalité Émergente par Auto-Organisation Coopérative. Habilitation à diriger des recherches, University of Paul Sabatier, Toulouse, France.

[3] Mefteh, W., Migeon, F., Gleizes, M-P. and Gargouri, F. 2012. Simulation Based Design. In 2nd International Conference on Information Technology and e-Services. 24-26 Mars 2012. Sousse, Tunisia.

[4] Cynthia, N and Gregory, M. 2009. A Survey of Various Agent Based Modeling Platforms. Journal of Artificial Societies and Social Simulation. ISSN : 1460-7425, Vol : 12, No 2.

[5] Ferber, J. 1995. Les systèmes multi agents. ,InterEditions.