International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 34224-34229
© Research India Publications. http://www.ripublication.com

Agent-Based Platforms Comparison

Nawfal NACIRI*& Mohamed TKIOUAT

Islamic Financial Engineering Laboratory (IFELab).
Studies and Research Laboratory in Applied Mathematics (LERMA).
Mohamed V University-Mohammadia school of Engineering.
Rabat, BP 765, Morocco.
naciri_nawfal@yahoo.fr& mohamedtkiouat@emi.ac.ma
**Corresponding author: naciri_nawfal@yahoo.fr*
+212661308963

Abstract
The multi-agent simulation consists in using a set of agent in interaction to reproduce the dynamics and evaluate phenomena that we try to simulate. It is considered as an alternative to the classical simulation, based on analytical models. To deploy and execute this set of agent, it must pass through the Agent based platforms. Agent based platform is a useful tool in simulating and exploring phenomena that consist of or can be thought of in terms of interactions between individual agents. There have been a number of other reviews of Agent based platforms. This paper, present platforms most used and most recognized in the field of multi-agent simulation namely: Agent builder, Zeus, DIMA, Jade, Madkit, Swarm, NetLogo and Janus. It regrouped its different methodologies, Comparison criteria's and classification techniques. The goal is to compares between eight platforms best known and most used, in order to classify them and identify the most efficient platforms.

Keywords: Complex systems;Multi-agent system; Multi-agent simulation; Agent based platforms; Comparison criteria.

1. Introduction

Although definitions of complexity vary greatly and are still a field of mines terminology, the scientific community agrees that the characteristics of complex systems are: a large number of autonomous entities, sensitivity and criticality to initial conditions, different organization levels, dynamic structures, emergent and self- organization properties [1]. Complexity exists in diver's fields: Economies, political systems, social networks, ecologies, and even our brains can be thought of as complex.

Understanding complex systems is done by their modeling. To do so, agent based platform appears the best answer to the needs of the complexity theory. Agent based platform is a useful tool in simulating and exploring phenomena that consist of or can be thought of in terms of interactions between individual agents[2].

The objective of this paper is to present and compare eight agent-based platforms: Agent builder, Zeus, Dima, Jade, Madkit, Swarm, NetLogo and Janus. We have holding eight most cited comparison criteria in the various reviews, namely: Simplicity, Popularity, Learnability, Performance, Stability, Robustness, Scalability and Security.

2. Agent based platforms comparison reviews

Agent based platforms are as diverse as the community of people who use them [3]. In the past few years, several seminal ABM surveys have emerged. They are a giant stride in the right direction, but current surveys generally are limited to four or five mainstay and characteristically or historically similar toolkits [4] [5] [6] [7]. There is also an exhaustive comparative table intended to capture many of the features that are important to Agent based platforms (see reference [8]).

The review by Najlis, Janssen & Parker [9] provides a good summary of capabilities of Swarm, Repast, CORMAS and another ABM platform not reviewed here, namely Ascape, which is no longer in development. [7**]** compared four agent-based tools: RePast, Swarm, Quicksilver, and VSEit. They rank the plat- forms by assigning the scores to represent the quality of the criteria of interest. Railsback, Lytinen & Jackson **[**6] wrote a review of agent based model platforms, focussing on epast, Swarm, MASON, and NetLogo. Other review by Matthew Berryman [2**]** witch focus on ABM toolkits for use for Complex Systems and Complex adaptive systems. [3] performed a survey of various agent- based platforms and characterized each platforms based on 5 characteristics: programming language, operating system, type of license, primary domain for which the toolkit is intended, and types of support available to the user. Bajracharya [10] wrote a review and comparison of three agent-based platforms based on a simple epidemiological model. Kravari [11] present the last comparative review of the most promising existing agent platforms that can be used.

First, we have holding eight most cited comparison criteria in the various reviews, namely: Simplicity, Popularity, Learnability, Performance, Stability, Robustness, Scalability and Security. Secondly, we applied the Bajracharya and Duboz [[10] classification technique and at the end, we arrive at the most effective agent-based platforms.

3. Presentation of Agent based platforms

In this part, we try to present platforms most used and most recognized in the field of multi-agent simulation namely: Agent builder, Zeus, Dima, Jade, Madkit, Swarm, NetLogo and Janus.

3. 1. Agent builder

Agent Builder is an integrated tool suite for constructing

intelligent software agents. Agents constructed using Agent Builder communicates using KQML and support the per-formatives defined for KQML. In addition, Agent Builder allows the developer to define new inter-agent communications commands that suit his particular needs.
Agent Builder consists of two major components: the Toolkit and the Run-Time System[12].
The Agent Builder Toolkit includes tools for managing the agent-based software development process, analyzing the domain of agent operations, designing and developing networks of communicating agents, defining behaviors of individual agents, and debugging and testing agent software
The Run-Time System includes an agent engine that provides an environment for execution of agent software.

3.2. Zeus

ZEUS is an integrated environment that uses the Role Modeling methodology for the quick construction of applications based on collaborative agents. It was developed by the Agent Research Program of British Telecom Intelligent System Research Laboratory.
The concepts of the platform ZEUS is based on: Agents, Their goals, Their spots (what agents should realize to achieve their goals), The facts (what agents consider to be true).
A ZEUS agent has a definition, it belongs to an organization, it receives and modifies its environment, it has a life cycle and it is based on an interaction protocol. The architecture of the ZEUS agents includes a mailbox, a message handler whose main role is to analyze the messages mailbox, a planner spots, a motor of coordination, an execution controller whose role is the management of active spots [13].
The hypotheses on ZEUS agents must be directed by goals, be able to plan, be honest with the other agents, may have several goals to satisfy simultaneously and can engage in several tasks at once[21].
The generic ZEUS agent consists of three inner layers required for the definition of agent as an autonomous entity with its skills and beliefs, the organization relations agent with other agents, and the coordination.
This generic agent contains two additional layers, one for communication with other agents, and the other is responsible for interaction with the environment.

3.3 DIMA

DIMA is a Java multi-agent platform built as an extension of Object-Oriented Programming (OOP). It provides generic and modular agent architecture with several facilities (lifecycle management, message passing, distribution...), and allows high heterogeneity in agent architectures (reactive, deliberative and hybrid), and various customizations.
DIMA is an attempt to provide answers to the basic questions on how to build a bridge between: 1) the implementation and modeling requirements of MAS and 2) the implementation and modeling facilities and techniques provided by OOP[14].

3. 4. JADE

JADE (Java Agent Development) is a software framework to make easier the development of agent applications in compliance with the FIPA specifications for interoperable intelligent MASs. The goal of JADE is to simplify development while ensuring standard compliance through a comprehensive set of system services and agents[15].
JADE offers the following list of features to the agent programmer[16]:

- FIPA-compliant Agent Platform, which includes the AMS (Agent Management System), the DF (Directory Facilitator), and the ACC (Agent Communication Channel). All these three agents are automatically activated at the agent platform start-up;
- Distributed agent platform. The agent platform can be split on several hosts (provided that there is no firewall between them). Only one Java application, and therefore only one Java Virtual Machine, is executed on each host. Agents are implemented as one Java thread and Java events are used for effective and lightweight communication between agents on the same host. Parallel tasks can be still executed by one agent, and JADE schedules these tasks in a more efficient (and even simpler for the skilled programmer) way than the Java Virtual Machine does for threads;
- A number of FIPA-compliant DFs (Directory Facilitator) can be started at run time in order to implement multi-domain applications, where the notion of domain is a logical one as described in FIPA97 Part 1;
- Programming interface to simplify registration of agent services with one, or more, domains (i.e. Directory Facilitator);
- Transport mechanism and interface to send/receive messages to/from other agents;
- FIPA97-compliant IIOP protocol to connect different agent platforms;
- Light-weight transport of ACL messages inside the same agent platform, as messages are transferred encoded as Java objects, rather than strings, in order to avoid marshalling and unmarshalling procedures. When sender or receiver does not belong to the same platform, the message is automatically converted to /from the FIPA compliant string format. In this way, this conversion is hidden to the agent implementers that only need to deal with the same class of Java object;
- Library of FIPA interaction protocols ready to be used;
- Automatic registration of agents with the AMS;
- FIPA-compliant naming service: at start-up agents obtain their GUID (Globally Unique Identifier) from the platform;
- Graphical user interface to manage several agents and agent platforms from the same agent. The activity of each platform can be monitored and logged

3.5. MadKit

MadKit is a Java multi-agent platform built upon an organizational model. It is developed by Gutknecht and Ferber. Contrarily to the other platforms presented in this work, MadKit is mostly a MAS runtime engine, using an

agent micro-kernel. The underlying organizational model is namedAalaadin
The MADKIT platform architecture is rooted in the AGR (agent-group-role) model developed in the context of the AALAADIN project.[17].

The MADKIT platform is built with three main components:

- *The MADKIT micro-kernel:* The MADKIT micro-kernel is a small (less than 40 Kb) and optimized agent kernel. The term "micro-kernel" is intentionally used as a reference to the role of micro-kernels in the domain of engineering.
- *Agentification:* The generic agent is MADKIT is a class defining basic lifecycle:
  - Control and life-cycle. The main agent class in MADKIT defines primitives related to message passing, group and role management,
  - Communication is achieved through asynchronous message passing, with primitives to directly send a message to another agent represented by its Agent Address.
  - Group and roles actions and requests are defined at action level. The agent developer is completely free to define the agent behavior
- *Graphic component model:* MADKIT graphic model is based on independent graphic components, using the Java Beans specification in the standard version.

3.6. Swarm

SWARM is a multi-agent software platform for the simulation of complex adaptive systems. The basic unit of a SWARM simulation is the agent. An agent is any actor in a system, any entity that can generate events that affect itself and other agents[18]. Simulations consist of groups of many interacting agents. For example, an ecosystem simulation could consist of agents representing coyotes, rabbits, and carrots. In an economic simulation, agents could be companies, stockbrokers, shareholders, and a central bank. Simulation of discrete interactions between agents stands in contrast to continuous system simulations, where simulated phenomena are quantities in a system of coupled equations.
The fundamental component that organizes the agents of a Swarm model is an object called a ``swarm.'' A swarm is a collection of agents with a schedule of events over those agents[19].
In SWARM, agents can themselves own swarms, models that an agent builds for itself to understand its own world. For example, in an economic simulation of a swarm of companies the researcher might be interested in the theory each company has of its competitors' actions. To model this in the Swarm system, the model builder would give each company-agent its own swarm: these private swarms implement each company's model of the world.

3.7. NetLogo

NETLOGO is a multi-agent programming language and modeling environment for simulating complex natural, economic and social phenomena. It is particularly well suited for modeling complex systems evolving over time. Modelers can give instructions to hundreds or thousands of independent "agents" all operating concurrently, in order to explore connections between micro-level behaviors of individuals and macro-level patterns that emerge from their interactions[20].
Historically, NetLogo is the next generation of the series of multi-agent modeling languages including StarLogo. NetLogo it written in Java so it can run on all major computing platforms.

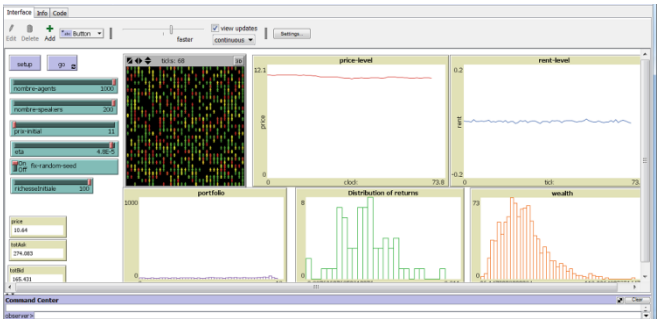

Figure 1: NetLogo's user interface, with model of Zero-intelligence trading within a double-auction order-driven stock market.

Figure 1, shows NETLOGO's user interface after opening and running a model of Zero-intelligence trading within a double-auction order-driven stock market. On the right is the graphics window, in which the "world" of the model is made visible. In the model shown, the turtles represent diffusing particles. They move randomly. When the model begins, there is a single green patch in the center. When a particle encounters a green patch, it "sticks" and turns green itself. Over time, a beautiful, branching aggregate emerges. On the left are model controls. In this model, they include:

- Buttons for controlling the model. "Setup" initializes the model and "Go" makes it run;
- Sliders that control model parameters. For example, the "num-particles" slider controls the number of particles that build the aggregate.

Agents can be inspected and altered and the code for the model can be changed without restarting the simulation[22]. At the bottom of the Interface tab is the "Command Center, " in which NETLOGO commands can be issued, even while the model is running. The other tabs are:

- Information, where documentation on the model is found. This typically explains the rules behind the model and suggests experiments for the reader to try.
- Procedures, where the actual code for the model is stored. A well-written model includes comments in the code explaining how it works.
- Errors (normally disabled), where any incorrect code can be viewed and fixed.

3.8. Janus

Janus is a multi-agent platform that was designed to deal with the implementation and deployment of holonic and multiagent systems. It is written in Java 1.5 and based on the CRIO organizational metamodel. Its key focus is that it supports the implementation of the concepts of role and organisation as first-class entities[23]

Janus was designed to facilitate the transition between design and implementation phase. It thus provides a direct implementation of the five key concepts used in the design phase[24]: organization, group, role, holon and capacity:

- *Organization*: The organization is implemented as a first-class entity, which includes a set of roles classes. An organization can be instantiated in the form of groups.
- *Group*: group contains a set of instances of different classes of roles associated with the organization, which it implements.
- *Role*: A role is local to a group, and provides holons playing the role the means to communicate with other group members.
- *Holon*: An agent is represented by an atomic holon (a non-composed one). Janus defines two main types of holon: *HeavyHolon* and *LightHolon*. A *HeavyHolon* has its own execution resource (one thread per holon), and can therefore op- erate independently. The *LightHolon* is associated with synchronous execution mechanisms and it is very useful to develop multiagent-based simulation.
- *Capacity*: The notion of capacity enables the representation of holon competences. Each holon has, since its creation, a set of basic skills, including the ability to play roles (and therefore communicate), to obtain information on existing organizations and groups within the platform, create other holons, and obtain new capacities.

4. Agent Based Platforms Classification

4.1 Comparison criteria

We have holding eight most cited comparison criteria in the various reviews, namely: simplicity, popularity, learnability, performance, stability, robustness, scalability and security. For these criteria, we take the same definitions as given in [11]**:** Simplicity indicates how simple and understandable the underlying platform's mechanisms are and how easy it is for a developer to use that platform. Learnability refers to whether the platform's mechanisms are familiar to the developer community or not. Scalability refers to a potential critical issue; whether the platform can handle varying problem sizes or not. Performance, is about how efficient is the platform with respect to space and time operations.Stability, is about how stable is the platform in case of long-term execution whereas. Robustness is about how tolerant it is in case of breakdowns. Security, refers to protocols and mechanisms that protect access to the platform

4.2 Classification

To classify these platforms, we applied the Bajracharya and Duboz [10] classification technique and at the end, we attain at the most effective agent-based platforms.

Following are the rating of platforms:

- Very high: platforms is highly suitable with respect to given criteria;
- Good: platform is suitable with respect to given criteria,
- Average: platform is averagely suitable with respect to given criteria,
- Low: platform is least suitable with respect to given criteria,

4.3 Results

The table below gives the rating of agent based platforms based on the features:

Table 1: Rating of agent based platforms based on the features

Comparaison criteria	Rank	W=n-r+1	Normalized Weight (wi)	AGENT BUILDER	ZEUS	DIMA	JADE	MADKIT	SWARM	NETLOGO	JANUS
Simplicity	1	8	0,222	Good	Good	Good	Good	Good	Average	Good	Good
Popularity	2	7	0,194	Average	Average	Average	High	Average	Average	High	Average
Learnability	3	6	0,167	Good	Good	Average	High	Average	Average	High	Good
Performance	4	5	0,139	Good	Good	Average	Good	Low	Low	Good	Good
Stability	5	4	0,111	Good	Good	Good	High	Good	Average	Good	Good
Robustness	6	3	0,083	Hight	Good	Good	High	Good	Good	Avergae	Good
Scalability	7	2	0,056	Good	Good	Good	Good	Good	Low	Good	Good
Security	8	1	0,028	Low	Low	Good	High	Good	Low	Avergae	Low

Each criteria is assigned a weight factor depending upon the rating of those criteria (formula 1). This allows us to calculate normalized weight: wi using formula (2)

$$f= \begin{array}{|l|l|} \hline \text{Very high} & 1 \\ \hline \text{Good} & 0{,}8 \\ \hline \text{Average} & 0{,}6 \\ \hline \text{Low} & 0{,}4 \\ \hline \end{array} \quad (1)$$

$$w_i = \frac{n - r_i + 1}{\sum_{k=1}^{k=10} n - r_k + 1} \quad (2)$$

The table below gives the normalized weights:

Table 2: Normalized weights

Comparaison criteria	Rank	W=n-r+1	Normalized Weight (wi)	AGENT BUILDER	ZEUS	DIMA	JADE	MADKIT	SWARM	NETLOGO	JANUS
Simplicity	1	8	0,222	0,8	0,8	0,6	0,8	0,4	0,4	1,0	0,8
Popularity	2	7	0,194	0,6	0,6	0,6	1,0	0,6	0,6	1,0	0,6
Learnability	3	6	0,167	0,8	0,8	0,6	1,0	0,6	0,6	1,0	0,8
Performance	4	5	0,139	0,8	0,8	0,8	0,8	0,8	0,6	0,8	0,8
Stability	5	4	0,111	0,8	0,8	0,8	1,0	0,8	0,6	0,8	0,8
Robustness	6	3	0,083	1,0	0,8	0,8	0,8	0,8	0,8	0,6	0,8
Scalability	7	2	0,056	0,8	0,8	0,8	0,8	0,8	0,4	0,8	0,8
Security	8	1	0,028	0,4	0,4	0,8	0,8	0,8	0,4	0,6	0,4

With the equation below (formula 3), we will calculate cumulative weightage:

$$W = \sum_{i=1}^{i=10} (w_i \times f) / (\sum_{i=1}^{i=10} w_i) \quad (3)$$

The table below gives the cumulative weightage calculation:

Table 3: cumulative weightage calculation

Comparaison criteria	Rank	W=n-r+1	Normalized Weight (wi)	AGENT BUILDER	ZEUS	DIMA	JADE	MADKIT	SWARM	NETLOGO	JANUS
Simplicity	1	8	0,222	0,1778	0,1778	0,1333	0,1778	0,0889	0,0889	0,2222	0,1778
Popularity	2	7	0,194	0,1167	0,1167	0,1167	0,1944	0,1167	0,1167	0,1944	0,1167
Learnability	3	6	0,167	0,1333	0,1333	0,1000	0,1667	0,1000	0,1000	0,1667	0,1333
Performance	4	5	0,139	0,1111	0,1111	0,1111	0,1111	0,1111	0,0833	0,1111	0,1111
Stability	5	4	0,111	0,0889	0,0889	0,0889	0,1111	0,0889	0,0667	0,0889	0,0889
Robustness	6	3	0,083	0,0833	0,0667	0,0667	0,0667	0,0667	0,0667	0,0500	0,0667
Scalability	7	2	0,056	0,0444	0,0444	0,0444	0,0444	0,0444	0,0222	0,0444	0,0444
Security	8	1	0,028	0,0111	0,0111	0,0222	0,0222	0,0222	0,0111	0,0167	0,0111
Total sum		36	1	0,7667	0,7500	0,6833	0,8944	0,6389	0,5556	0,8944	0,7500

According to the table above, we find that both platform ranks first, JADE and NetLogo. However, if we take into consideration their field of application (even if difficult to limit its), it appears that Netlogo is strongly recommended. NetLogo contains models that are compatible with its paradigm of short-term, local interaction of agents and a grid environment and not extremely complex. Also for its prototyping implement models that may later in lower-level platforms.

5. Conclusion

Agent-based modeling and simulation has become increasingly popular as a modeling approach in various disciplinary fields like computer science, economics, businesses problems, social science, earth science, ecology, etc. Agent-based platforms are the framework developing the models in an easier way to help users to understand and develop the model of complex system in a new approach.

In this article, we have presented eight best known and most used Agent based platforms, in order to classify them and identify the most efficient platforms. We figured out that NetLogo remains the most popular platform and itis strongly recommended. It is particularly well suited for modeling complex systems evolving over time.

Our qualitative comparison and classification, proposed an overview of the different alternatives. Researchers, who are study agent based platforms, can study the above presented tables (1, 2, and 3) in order to find the intersection that better describes their needs.

6. References

1. Naciri & Tkiouat 2014: Nawfal NACIRI & Mohamed TKIOUAT. IEEE Publication. 2014 Second World Conference on Complex Systems (WCCS) 2014.
2. Matthew 2008: Matthew Berryman: « Review of Software Platforms for Agent Based Models". Land Operations Division. Defence Science and Technology Organisation. Edinburgh South Australia 5111 Australia.)
3. Nikolai and Madey (2009): Cynthia Nikolai and Gregory Madey"Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. Journal of Artificial Societies and Social Simulation vol. 12, no. 2 2. 2009
4. Serenko and Detlor, 2002: Serenko A and Detlor B "Agent Toolkits: A General Overview of The Market and an Assessment of Instructor Satisfaction with Utilizing Toolkits in the Classroom". Working Paper 455, 2002
5. Castle et al, 2006: Castle C and Crooks A Principles and Concepts of Agent-Based Modelling for Developing Geospatial Simulations. Working paper 110, Centre for Advanced Spatial Analysis, University College London, September 2006.
6. Railsback et al, 2006: Railsback, S, Lytinen, S and Jackson, S (2006) Agent-Based Simulation Platforms: Review And Development Recommendations. Simulation,
7. Tobias et al, 2004: Tobias R and Hofmann C "Evaluation of free Java-libraries for social-scientific agent based simulation". Journal of Artificial Societies and Social Simulation 7 (1) 6, 2004
8. W1: http: //en.wikipedia.org/wiki/Comparison_of_agent-based_modeling_software Accessed 28/04/2015
9. Najlis et al 2001: Najlis, R., Janssen, M. A. & Parker, D. C. (2001) Software tools and communication issues, in Proc. Agent-Based Models of Land-Use and Land-Cover Change Workshop, pp. 17–30.
10. Bajracharya et al, 2012: KishojBajracharya, Raphael Duboz, 2012: "Comparison of three agent-based platforms on the basis of a simple epidemiological model (WIP)" Asian Institute of Technology PathumThani, Thailand
11. Kravari et al, 2015: Kalliopi Kravari and Nick Bassiliades "A Survey of Agent Platforms" Journal of Artificial Societies and Social Simulation 18 (1) 11.2015".
12. AgentBuilder 2000., An Integrated Toolkit for Constructing Intelligent Software Agents, AgentBuilder, Reference Manual, April 2000
13. W3: http: //www.labs.bt.com/projects/agents/zeus Accessed 09/06/2014
14. Guessoum et al, 2014: Z. Guessoum et J-P. Briot « DIMA », OASIS, Laboratoire LIP6, Univ. Paris 6 http: //www-poleia.lip6.fr/~guessoum/dima.html
15. Bellifemine et al, 2000: Bellifemine F., Giovani C., Tiziana T. Rimassa G., "Jade Programmer's Guide" Jade version 2.6 (http: //sharon.cselt.it/projects/jade/), 2000.
16. Jade 2000: Plate-forme JADE: Java Agent Development Framework, 2000. http: //jade.cselt.it/
17. Gutknecht et al 2000 ; Gutknecht, Ferber ; Michel: « MADKIT: Une plate-forme multi-agents générique»
18. W2: www.swarm.org. Accessed 22/12/2014
19. Minar et al., 1996: Nelson Minar: The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations
20. Seth, 2004: Seth Tisue: ''NetLogo: Design and implementation of a multi-agent modeling environment '' Uri Wilensky
21. Nwana 1999: ZEUS: A Toolkit for Building Distributed Multi-Agent Systems
22. Dang Kim Dung, 2008: étude comparative sur les plateformes de simulation à base des systèmes multi-agents NetLogo et GAMA

23. Gaud et al, 2009: Nicolas Gaud, StéphaneGalland, Vincent Hilaire, and AbderrafiaaKoukam: An Organizational Platform for Holonic and Multiagent Systems. Multiagent Systems Group, System and Transport Laboratory University of Technology of Belfort Montb´ eliard 90010 Belfort cedex, France.2009
24. Galland2013: StéphaneGalland. "Multiagent-based Simulation: Organizational and Agent Modeling, Janus Platform." DATASIM SummerSchool 2013.Universit´e de Technologie de Belfort-Montbéliard