



**HAL**  
open science

## Online analysis and visualization of agent based models

Arnaud Grignard, Alexis Drogoul, Jean-Daniel Zucker

► **To cite this version:**

Arnaud Grignard, Alexis Drogoul, Jean-Daniel Zucker. Online analysis and visualization of agent based models. ICCSA, Jun 2013, Ho Chi Minh City, Vietnam. pp.662-672. hal-00876569

**HAL Id: hal-00876569**

**<https://hal.science/hal-00876569>**

Submitted on 25 Oct 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Online analysis and visualization of agent based models

Arnaud Grignard<sup>1</sup>, Alexis Drogoul<sup>1,2</sup>, Jean-Daniel Zucker<sup>1,2</sup>

<sup>1</sup> UMMISCO/UPMC, Paris, France

<sup>2</sup> MSI-UMMISCO/IRD, Hanoi, Vietnam

**Abstract.** Agent-based modeling is used to study many kind of complex systems in different fields such as biology, ecology, or sociology. Visualization of the execution of a such complex systems is crucial in the capacity to apprehend its dynamics. The ever increasing complexification of requirements asked by the modeller has highlighted the need for more powerful tools than the existing ones to represent, visualize and interact with a simulation and extract data online to discover imperceptible dynamics at different spatio-temporal scales. In this article we present our research in advanced visualization and online data analysis developed in GAMA an agent-based, spatially explicit, modeling and simulation platform.

**Keywords:** Agent Based Modeling, online analysis, visualization, interaction, 3D, complex systems

## 1 Introduction

Agent-based modeling has undergone an incredible, albeit slow, evolution over the last 30 years. From simple to apprehend abstractions that merely served to underline intriguing dynamics [15] under the KISS (Keep It Simple, Stupid) paradigm, it has progressively become the technology of choice for designing large, complex models that follow the KIDS (Keep It Descriptive, Stupid) approach advocated by [7]. In this approach, a model of a real system is built to provide modelers or end-users an experimental playground that allows altering any component of the model and explore the dynamics that result from these modifications. Instead of providing an analytic vision of the system, agent-based modeling becomes, under this paradigm, a heuristic method to understand, by simulation, the possible evolutions of the described system. This evolution has made a number of new requirements emerge, among which: (a) the possibility to set up a complete experimental environments around models, based on rich visual feedback and interactivity; (b) the possibility to extract interesting dynamics in the simulations thanks to the use of analysis tools, data-mining and high-performance computing techniques [4].

As a matter of fact, it is now widely recognized that achieving a comprehensive understanding of a complex Agent Based Model (ABM) through an experimental approach has to rely on: (1) dynamic, realistic and intelligible visualization of its evolution under different setups, (2) the ability to easily (re)define this visualization, ideally in a an interactive way, at different spatial and temporal scales or using different points of view, (3) the ability to extract abstract properties and knowledge from the dynamics generated by the simulation and to visualize them as well in real-time, (4) the ability to interact with such visualizations in a natural way in order to modify the model itself in an interactive design approach.

Nowadays, these different requirements are hot research topics [1], but have led, so far, to few usable implementations. Some graphical languages dedicated to the visualization of agent-based models have been proposed in different ABM platforms, but they are often too simple or too complicated to be used by non computer scientists. There is still no possibility, in any of these platforms, to have different points of view on a given model, to confront them and make them evolve during simulations. In addition, statistical or data-mining tools are mostly available as independent tools that are used to analyze the outputs of simulations after and not during their executions. Building abstractions online and using them to facilitate the understanding of the dynamics of a model through advanced visualization techniques, is still out of reach in existing ABM tools.

After giving an overview of the current practices and related work in agent-based model visualization we propose an on-going research on a language for model visualization and online tools to analyze and abstract data (based on graph representation) online. Most of the tools proposed here have already been implemented and tested in GAMA version 1.5 [17], others are at the stage of design and are implemented in the soon to be released in version 1.6.

## 2 Related Work

In already existing popular platforms (NetLogo, Repast, Mason, Swarm, Anylogic), languages for visualization do exist [14] but most of them lack tools for building, observing and interacting with models. In effect, Most of them offer basic tools such as 2D animation window representing agents with dynamic color, inspector, model parameter display and chart output. On top of it, standardization emerges and help the user to build a meaningful model representation that follows design technique guidelines which are often based on a non hierarchical categorization of ABM models such as conventional ABM visualization (natural phenomena or mathematical representations), structured ABM visualization (aggregation of many agents creating an emergent pattern), unstructured ABM visualization (irregular distribution of agents) [6]. Unfortunately, this description only stay focused on the agent representation and do not propose any abstraction and multi-level representation or only by using ad-hoc solutions and post simulation treatment that are not yet standardized.

In existing platforms outputting data in a file and data recording can easily be scheduled like any other action but the analysis is done offline and most of the time when the simulation is finished. Only few platform propose built-in analysis tools that run online and serve as feedback or indicator during the simulation by representing macro-behaviour and using graph analysis or data clustering to build groups of agents [11]. However, recent work on the multi-level approach [9] propose solution to consider entities at several levels of scales through an analytical framework and an automated description of group dynamics by the use of statistical based tools [3].

To conclude visualization is often seen as a cosmetic aspect of the model, and rarely considered as an essential part of the modelers abstraction and comprehension work. Visualization is not used as an experimental approach but rather as a final result. We present in this paper a new way to represent model and simulation as well as tools to extract abstract data online.

### 3 New way to visualize agent-based model

Building a visualization is like building a visualization model on the model itself and the construction of this "visual" model requires its own language and need to be independent of the execution of the model to define different representations of a given model without modifying the model itself. A dedicated language for visualization allows to describe graphical primitive (from simple geometry to more complex objects such as graph, dendrogram or 3d object) to construct a sensitive representation of the model, allowing feedback on the model and its dynamic from the visualization, and offering modellers a simple way to declare it. In addition, a complete understanding of a model cannot be dissociated from (1) basic visualization concepts, such as: overview, zoom and filter, details on demand [16]. (2) basic interaction concepts to help the user to adjust the model during the simulation through the use of probes, dynamic filtering, highlighted details and different level of details representations. Above the model representation, a model is included in a scene with at least one point of view, a description of phenomenon such as lighting, shading, transparency with a given resolution both in time and space [12].

Starting from geometric representation of 2D and 3D to graph visualization, GIS data integration and multi-level representation, the techniques described below are implemented in the platform Gama, whose default display, based on the Java2D API, has shown some limits when dealing with large-scale models and realistic rendering. The limits address by JAVA2D was first the impossibility to define and represent 3D geometry but it also showed limits in term of real time rendering when dealing with large models. To address this limitation OpenGL display has been integrated in GAMA 1.5. This not only helps the user to build aesthetic visualization by offering most of the common metaphors for visualizing complex systems but also offers much more in term of graphical performance.

### 3.1 From Data to Form

There are many ways to visualize and distinguish data elements: size, value, texture, color, orientation and shape. For each agent the value of one of its attributes can be represented in different ways such as a text, a color, a 2D or 3D geometry. In many cases the shape of the agent can be used to represent one or several values of its attributes. Gama handles in a seamless way vector data that are integrated as the simulation environment as an input but also as an output to store resulting experiment [17]. Graphic operator applied on vector data are used to compute distance, surface, neighbourhood or even more complicated operation such as shortest-path computation, geometry intersection, union, difference and many other spatial algorithms. Those graphical operators are also used for the agent aspect definition through pre-built graphical primitive described in figure 1.














|    | Point   | Line  | Circle  | Square  | Polygon   | Bitmap  | GIS   |
|----|---|---|---|---|---|---|---|
| 2D | +   |    |    |    |    |   |    |
|    | Sphere  | Plan  | Cylinder  | Cube  | Polyhedron  | 3D Object   | 3D GIS  |
| 3D |  |  |  |  |  |  |  |

Fig. 1. Common graphical primitives used in ABM visualization.

It is straightforward to define several aspects for a given agent. Figure 2 shows different displays applied on the same 6x6 Grid where each agent (a cell) has a specific value represented by 4 different aspects (text, square, circle, cylinder).

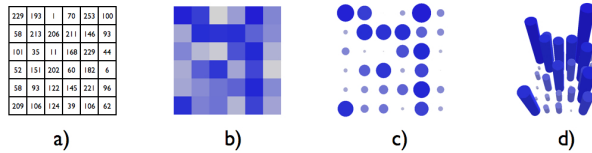


Fig. 2. Different aspects of the same model (6x6 Grid). a) value is displayed as a text. b) value is displayed as a colored square. c) value is displayed as a circle with a variable radius. d) value is displayed as an cylinder with a variable elevation.

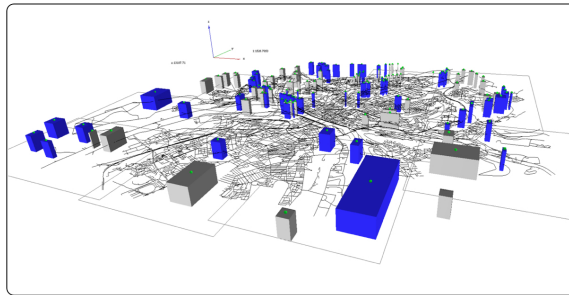
Once agent aspect have been defined, a 3D scene displays all the agents present in the model with their respective aspect on different layer. By using intuitive 3D navigation any location in the model can be reach to observe the model from different point of view. In addition, any agent can be selected, inspected and modified through the user interface. The layer control is a smart way to make visible layer or not in the 3 dimensions or to superpose them for a better readability. Finally when a user wants to share his work or record a given step of the simulation with a given point of view the snapshot tool is here to capture the model at a given spatio-temporal step. All the features described in figure 3 has been implemented in GAMA.



**Fig. 3.** Interaction tools

### 3.2 Integration with GIS Data

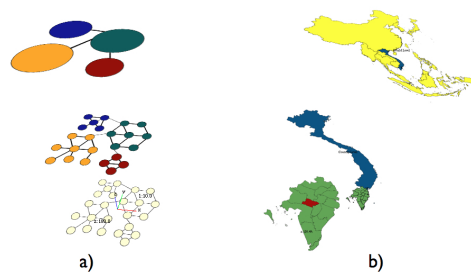
A large scale of geographical vector datasets are now available and used to face problem that integrate spatial dimension. Using this type of data is now often required to make the simulations closer to the field situation and it allows to use tools, like spatial analysis, coming from Geographic Information Systems (GIS) to manage these data. GIS technology first aim is to represent map and it cannot provide adequate performance with very large datasets on a huge amount of iteration when applied in dynamic simulation where spatial analysis is necessary [5]. To face this issue, in Gama, any geographical object is transformed in an agent with its own internal state and behavior and it can go further by easily converting any agent to a geographical object by spatializing it. Some of the already existing platform support geographical data [5] but GAMA offers much more in term of GIS services and in operations on geographical vector data [17]. We use functions from GeoTools to import and export data and Java Topology Suite (JTS) for data manipulation. As the integration of GIS data is seamless and straightforward it is very easy to develop model mixing data coming from GIS and built-in agent in a 3D environment. Figure 4 is an illustration of the mixing of GIS data with the integration of building agent.



**Fig. 4.** GIS integration in ABM

### 3.3 Multi-Level Modeling

Complex system is by definition mixing different entities at different levels of organization and the visualization of those different level is crucial for the good understanding. Most of the already existing approaches are called emergentist approach where only the lower model is present in the model and where the higher level is only an observation of the model but is not reified in the model itself as it emerges during the simulation. Recently new research has been produced in considering entities at different levels [9]. This approach focus on how to describe the articulation and influence of different levels. In this kind of approach, a powerful visualization toolkit is, among other statistic tools, one of the more intuitive way to understand, describe and perceive the multi-level paradigm. Gama meta-model has been design to take in account simultaneously several levels in the same model [18]. Thus, we can easily represent multi-level model using multi-layer rendering where layers can be then placed on different z value to represent different level of organisations as shown in figure 5.



**Fig. 5.** Multi-Level 3D representation. a) Simple graph, clustered graph and macro graph. b) district, country and world level.

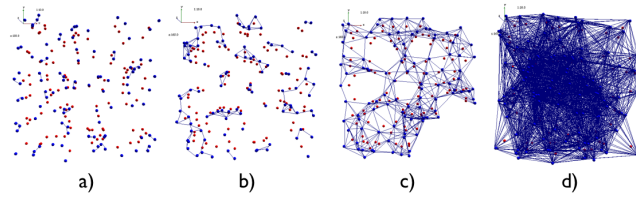
## 4 MODAVI - Multi-Level Online Data Analysis and Visualization

Agent-Based Simulations generate massive loads of data, which are usually analysed at the end of simulations. In large scale model it's not always easy to have a clear idea of all the interactions occurring between agent. Interaction analysis is part of the different frameworks to explore a complex systems where macro-level dynamics and structure are caused by the interactions of a relatively high number of agents. MODAVI is an exploratory tool that addresses the problem of online analysis of an ABM simulation using the new available graphic feature developed in GAMA such as graph agentification, multi-level modeling and 3D visualization, to handle online data abstraction. It proposes to build an online network abstraction to represent interaction that can appear between entities (i.e gene regulatory network, infectious contact, betweenness individual in a population, etc). This process is related to data stream mining which is the process of extracting knowledge structures from continuous, rapid data records. The MODAVI approach supports representing the interdependence between various interacting entities to be observed as abstract agent. Such an online mining tool may be used to explore any kind of interaction between agents [13]. The visualization of the results and their interpretation plays a key role in the understanding of a simulation and it answers to questions in a manner that is not possible with empirical observation and experimentation. MODAVI is at the boundary between visualization and analysis and can be used to describe and represent indicator to both assess and explore a model. It goes beyond model visualization as the information displayed is the result of mining algorithms such as clustering that are performed on the stream of data from the ABM simulation. This tool provides new exploration capabilities and strategies to generate, analyze and visualize a large amount of alternative simulation experiments. MODAVI is based on two concepts developed in Gama, the proxy graph pattern that creates an interaction graph from a population of agent and a macro graph generator that creates a macroscopic description of the interaction graph.

### 4.1 Proxy Agent Pattern

From a given population of agent, the proxy graph pattern creates an interaction graph. In a proxy graph, a proxy node is an agent that mirrors a given agent. By creating proxy node we insure that the target population is not affected by the creation of the graph. The link between the agent and the proxy node is dynamic insuring the mapping between the proxy node and its target node at any time of the simulation. Two proxy nodes are connected together if their distance is smaller than a given threshold as shown in figure 6. This proxy agent provides more in term of representation in the sense that it enables to manipulate and thus altering the representation of the proxy agent without modifying the target agent.





**Fig. 6.** From a population of  $n$  agents in red, a proxy graph is created where proxy node in blue are connected with other proxy node if their distance between them is smaller than a given threshold. a) threshold=5 - b) threshold = 10 - c) threshold= 20 - d) threshold= 50

## 4.2 Macro Graph Generation

From an already existing graph at a micro level, a macro graph that clusters groups of agents and summarizes the number of links between each class of agents is created online see figure 7. A macro node represents all the aggregated agents of a given population at each iteration. A macro node is represented by a sphere whose radius evolves according to the number of aggregated nodes. A macro edge linked a macro node representing (aggregating) all the agents A and a macro node representing (aggregating) all the agents B. A macro edge is the representation of the total number of edges between agent A and agent B. The macro edge is created thanks to a linkage function that measures the distance between two groups. We define here four different linkage function:

- Single linkage: the similarity of two clusters is the similarity of their most similar members.

$$D(X, Y) = \min_{x \in X, y \in Y} d(x, y)$$

- Complete linkage: the similarity of two clusters is the similarity of their most dissimilar members.

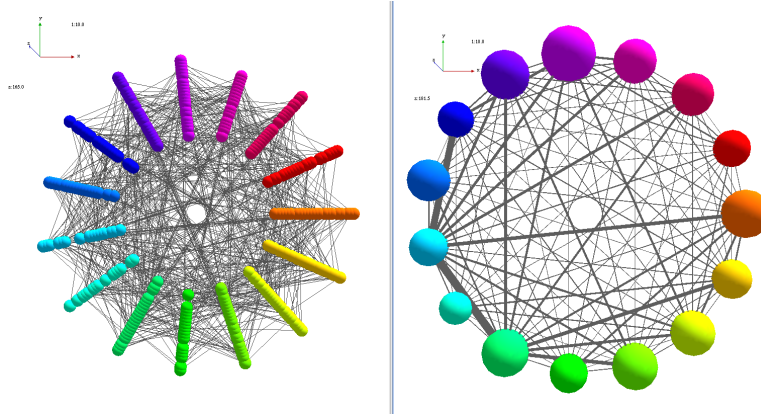
$$D(X, Y) = \max_{x \in X, y \in Y} d(x, y)$$

- Average linkage: the similarity of two clusters is computed as the average distance between objects from the first cluster and objects from the second cluster.

$$D(X, Y) = \frac{1}{N_X N_Y} \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} d(x_i, y_j)_{x_i \in X, y_j \in Y}$$

- Average group linkage: the similarity of two clusters is computed as the distance between the average values (the mean vectors or centroids) of the two clusters.

$$D(X, Y) = \rho(\bar{x}, \bar{y}); \bar{x} = \frac{1}{N_X} \sum_{i=1}^{N_X} x_i = \frac{1}{N_Y} \sum_{i=1}^{N_Y} y_i$$



**Fig. 7.** Micro Graph and the corresponding Macro Graph using an average linkage function. In the macro graph a node is the aggregation of all the agent in the micro graph with the same color, and an edge between macronode A and macronode B is the aggregation of all the edges between micronode a and micronode b.

## 5 Usability and Results

GAMA is being used as a decision-support tool for natural resource management in the MAELIA [8] project. This project aims at studying the social, economic and ecological impact of water management in the Adour-Garonne Basin (France) by the integration a huge amount of geographical data and models (from ecological models to human decision-making models). The model produces thus data that cannot be understood without advanced visualization features. In particular, to visualize various views of the same model with georeferenced or aggregated data and to display at the same time the areas with water lacks, the agricultural activities and the plant growth states with a spatial representation and time series of water flows aggregated at the level of the whole basin. All these information need separated displays to be understandable.

GAMA is also used as a decision-support tool for studying daily urban dynamic in the MIRO project [2]. The Miro model addresses the issue of sustainable cities. Therefore, improving urban accessibility merely results in increasing the traffic and its negative externalities, while reducing the accessibility of people to the city. Given real data the simulator is used to realise scenarios determined by geographers for quantifying service accessibility and identifying cities management strategies. Such simulation produce huge amount of data especially geolocalized ones. To sum up them we imagine 3D visualisation associating data, space and time such as proposed in time-geography domain. In spite of their importance spatio-temporal prism is a visualisation proposed by time-geography that have not any response in simulation domain. Proposing this kind of visualisation in the simulation will open new perspective of the geography domain.

## 6 Conclusion and Perspectives

As modeling complex systems is becoming more and more pervasive there is a growing need for online visualization tools that support making sense of simulations. To that extent, this paper focus on: (1) the setting of a graphical experimental environment based on high-level graphics and dedicated graphical language, (2) MODAVI, an approach that abstracts dynamics and provide interactive online analysis during the simulation. Most of the concepts described in this article have been implemented in GAMA while some others are still in under development. Besides those researches we also work on rendering optimization to achieve high performance rendering and to deal with real time application. Future work will includes a graphical language to modify the model – in a more intuitive way than with the code – will explore new way (e.g such as ones offered by web technology) to share simulation where a model could be played at runtime (online) or in a replay mode (offline) and where any agent could be inspected at any spatio-temporal scale during the simulation on different devices such as computers, or embedded devices. The next version of GAMA will (V. 1.6) also support exporting simulation in an open standard xml called COLADA. It will support exchanging digital assets and especially simulation and replay them on different architecture later on and share knowledge about model. Thus the whole dynamics is captured and one can replay the simulation with the ability to move in the scene and inspect any agent at any time step. In term of spatio-temporal scale we are working on spatio-temporal geo-visualization where different techniques have evolved over the years such as 2D Maps, Animation and SpaceTime Cubes where the time dimension is orthogonal to the surface of the map [10]. Finally, high-level Graphics library mixed with web technology open new way to develop participatory simulation where multiple humans control or design individual agents in the simulation or in new field such as crowdsourcing where a groups of humans, working together in parallel, produce a work impossible to achieve alone or serially.

## References

1. Allan, R.: Survey of agent based modelling and simulation tools. Science & Technology Facilities Council (2010)
2. Banos, A., Marilleau, N.: Improving individual accessibility to the city: an agent-based modelling approach. ECCS (2012)
3. Caillou, P., Gil-Quijano, J.: Simanalyzer: automated description of groups dynamics in agent-based simulations. pp. 1353–1354. AAMAS '12, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2012)
4. Chuffart, F., Dumoulin, N., Faure, T., Deffuant, G.: Simexplorer: Programming experimental designs on models and managing quality of modelling process. IJAEIS (2010)
5. Crooks, A.T., Castle, C.J.E.: Agent-Based Models of Geographical Systems. Springer Netherlands, Dordrecht (2012)
6. Daniel Kornhauser, U.W., Rand, W.: Design Guidelines for Agent Based Model Visualization. Journal of Artificial Societies and Social Simulation vol. 12, no. 2 12(2) (2009)
7. Edmonds, B., Moss, S.: From kiss to kids—an anti-simplistic modelling approach. Multi-Agent and Multi-Agent-Based Simulation pp. 130–144 (2005)
8. Gaudou, B., Al: The maelia multi-agent platform for integrated assessment of low-water management issues. MABS 2013, Multi-Agent-Based Simulation XIV - International Workshop (2013 (To appear))
9. Gil-Quijano, J., Louail, T., Hutzler, G.: From biological to urban cells: lessons from three multilevel agent-based models. Principles and Practice of Multi-Agent Systems pp. 620–635 (2012)
10. Kjellin, A., Pettersson, L.W., Seipel, S., Lind, M.: Evaluating 2d and 3d visualizations of spatiotemporal information. ACM Transactions on Applied Perception (TAP) 7(3), 19 (2010)
11. Lamarche-perrin, R., Demazeau, Y., Vincent, J.m.: How to Build the Best Macroscopic Description of your Multi-agent System ? pp. 1–18 (2013)
12. Navarro, L., Flacher, F., Corruble, V.: Dynamic Level of Detail for Large Scale. AAMAS '11: The 10th International Conference on Autonomous Agents and Multiagent Systems 2, 701–708 (2011)
13. Prifti, E., Zucker, J.D., Clement, K., Henegar, C.: FunNet: an integrative tool for exploring transcriptional interactions. Bioinformatics (Oxford, England) 24(22), 2636–2638 (2008)
14. Railsback, S.F., Lytinen, S.L., Jackson, S.K.: Agent-based Simulation Platforms: Review and Development Recommendations. Simulation pp. 609–623 (2006)
15. Schelling, T.C.: A process of residential segregation: neighborhood tipping. Racial discrimination in economic life 157, 174 (1972)
16. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. IEEE Symposium on Visual Languages (1996)
17. Taillandier, P., Amouroux, E., Drogoul, A.: GAMA : a simulation platform that integrates geographical information data , agent-based modeling and multi-scale control. Principles and Practice of Multi-Agent Systems and Writing (13th International Conference, PRIMA 2010, Kolkata, India, November 12-15, 2010), 67–74 (2010)
18. Vo, D.A., Drogoul, A., Zucker, J.D.: An operational meta-model for handling multiple scales in agent-based simulations. In: Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference on. pp. 1–6. IEEE (2012)