

A model-view/controller approach to support visualization and online data analysis of Agent-based simulations

Arnaud Grignard, Alexis Drogoul, Jean-Daniel Zucker

► **To cite this version:**

Arnaud Grignard, Alexis Drogoul, Jean-Daniel Zucker. A model-view/controller approach to support visualization and online data analysis of Agent-based simulations. Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2013, pp.233-236. 10.1109/RIVF.2013.6719899 . hal-00982756

HAL Id: hal-00982756

<https://hal.archives-ouvertes.fr/hal-00982756>

Submitted on 24 Apr 2014

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.

A Model-View/Controller approach to support visualization and online data analysis of Agent-Based Simulations

Arnaud Grignard ^{*}, Alexis Drogoul [†], Jean-Daniel Zucker [†]

^{*}UPMC / UMI 209 UMMISCO, Institut de la Francophonie pour l'Informatique (IFI), Ha Noi, Viet Nam

[†]IRD / UMI 209 UMMISCO, Institut de la Francophonie pour l'Informatique (IFI), Ha Noi, Viet Nam

Emails: agrignard@gmail.com, alexis.drogoul@gmail.com, jdzucker@gmail.com

Abstract—Agent-based Modeling is playing a key role in an increasing number of approaches addressing modeling complex systems. Historically, such models were focused on describing the system modelled dynamics but not the interaction or visualization of the model itself. The new requirements for high-level realistic visualization and online analysis tools of ABM simulations raise key issues that are yet unsolved: how to visualize interaction between entities and more generally abstracting the key information from the system dynamics. Instead of ad-hoc existing approaches that require reification in the model, we propose an approach dedicated to visualize and discover emerging dynamics from the model. In this approach, the execution of the simulation and its representation is separated and building a visualization is like building a visualization model on top of the reference model. The techniques presented here have been implemented in an Agent-based simulation platform, Gama, and illustrate the new perspective on visualization in Agent-based Modeling.

Keywords: Agent-based model, visualization, interaction, online analysis, 3D, complex systems.

I. INTRODUCTION

More and more models in the field of complex systems are based on the agent-based model (ABM) paradigm. ABM consists of the association of three basic elements: a set of agents, an environment in which the agents interact, and a set of rules and relationships that guide agent behaviour. Agent-based modeling has undergone several evolutions over the last 30 years. From KISS (Keep It Simple, Stupid) models to KIDS (Keep It Descriptive, Stupid) models [1], it has become the technology of choice for designing rich descriptive models of large-scale complex systems. This has given rise to a parallel increase in the requirements of modellers and users regarding the presentation of such models and how to extract relevant informations and interact with them. We believe that recent advance in the visualization could greatly benefit ABM domain. Indeed, visual information combined with interaction techniques help to acquire knowledge by allowing the user to see, examine and make sense of a large amounts of information at once by the use of a visual representation. Nowadays, these different topics are hot research topics in the ABM community [2] but have led to few usable implementations in the sense that many ad-hoc solution exist but no generic approach are available.

The objective of our work is to propose a generic framework to separate the visual model from the underlying reference model to offer different visual representations at different levels of abstraction and insure that the visualization of a simulation and the interaction on a simulation are independent processes that do not alter the reference model itself. In this approach a view is seen as a model on which once can represent, abstract, interact without changing the definition of the observed model.

We propose here an approach, inspired by the Model View Controller (MVC) pattern. The analogy with the MVC is used to focus on the fact that view and controller are separated from the reference model but are still using the same ABM paradigm. After describing our approach, we will use it as a tool to handle multi-scale online data analysis on networks and will apply it on an epidemiological model. All the results shown here have been successfully implemented in the GAMA (Gis & Agent-based Modelling Architecture) simulation platform [3] [4]

II. RELATED WORK

In ABM, the role of visualization is different depending on the community. Model developers and researchers tend to use simple but exact and accurate representations (Repast [5], Netlogo [6]), educational community uses more figurative ones (StarLogo [7]) and business and professional use more impressive representations (AnyLogic [8]). In already existing popular platforms, languages and tools for visualization do exist. Starting from simple 2D representation using bitmap data to high-level graphics library dealing with vector based graphics and 3D representation [9] most of them lack techniques for building, observing and interacting with models [10]. Those platforms stay focused on the agent representation and do not propose any abstraction and multi-level representation or only by using ad-hoc solutions, loose coupling or post-simulation treatment that are not yet standardized [11]. If some standardization categorization of ABM gives guidelines for a better agent-based model representation [12], only few platform propose built-in analysis tools that run online and serve as feedback or indicator during the simulation (e.g. using graph analysis or data clustering to build groups of agents [13]).

In most of the ABM simulation platforms, agents can have several representations of themselves in different topologies by the use of modular plug-in architecture that enables a strict separation between model specification, model execution, model visualization and data storage like in Repast Symphony [5] but those representations only focus on how to represent the reference model but are never seen as new model (in the sense of ABM model) on which the user can prospect.

As shown in Figure 1, in a classical ABM approach, beside the data that represent the inner behavior of the agent, an agent has its own representation (aspect). This aspect is most of the time dedicated to represent the agent by a given shape with a given size that can be related to a given parameters but no inner behavior can be added in the aspect part.

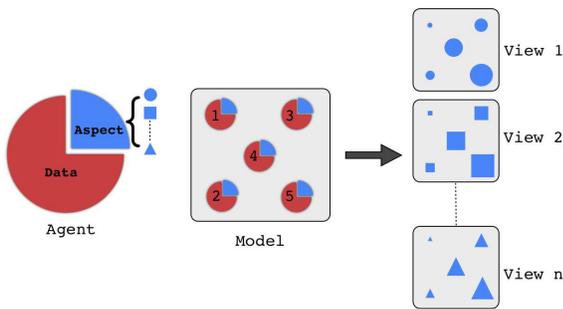


Fig. 1. In a classical ABM approach, an agent defines its aspects and can be displayed with different aspects on different topology in different views.

III. MATERIALS AND METHODS

The main goal of our approach is to separate the execution of the simulation with its representation and offering modellers a simple way to declare it. Visualization and interaction model are built on top of a reference model to easily represent abstract data coming from the simulation above the proper model representation. A dedicated language for visualization helps to construct a sensitive representation of the model, and allows feedback on the model and its dynamic from the visualization. In this mind building a visualization is like building a visualization model on the model itself where it becomes easy to isolate key agent of the reference model (proxy agent) to build new view model without altering the reference model. With this approach, firstly, the reference model is not aware of the model build on top of it and secondly, many models can exist on top of a reference model in order to represent it in different way by using successive level of abstraction as shown in Figure 2.

Model

A reference model is a model whose reference system is any real domain (complex system, biological, etc.). In that kind of model it is not natural to add representation or interaction element because those element do not represent entity or concept of the given domain but notion linked to other domain.

View/Controller

A View model is a visual and abstract model whose reference system is not any more a real domain but another ABM model (either a real model (reference model) or even another visual/abstract model). In this case view model agent can represent either agent of the referenced system, aggregation of agent or even totally abstract properties. A controller model supports the interaction with the user either to manipulate one or several view models or to manipulate the reference model itself. The interaction is handle by either one agent (providing general controls), or by several agent that support responses to those events to change their target.

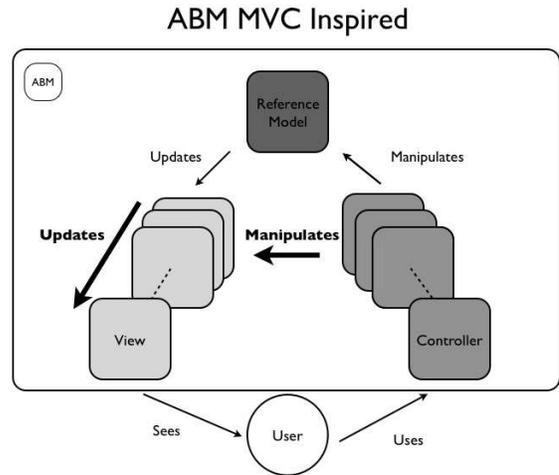


Fig. 2. In ABM MVC inspired approach, reference model, view model(s) and a controller(s) model are all seen as agent-based model. A view model represents a reference model or another view model on which it can request data. A controller can manipulate agents or group of the reference model or the view model(s).

GAML

GAML(GAMA Modeling Language) is the modeling language of the GAMA platform [14] [15]. This language, designed as a simple scripting language, has grown into a general language dedicated to the modeling and is here used to handle principles of visual, motion and interaction. GAML language is a text programming language and the visualization part has been inspired by Processing [16]. In Gama, primitive shapes (point, line, polygon) are manipulated through basic mathematical operations used to control, transform(translate,rotate,scale) and combine properties of visual elements to describe dynamics properties (size, texture, shape, orientation, color, text). Conditional structures are then used to control the flow of the program (relational, conditional, logical) and can finally be inserted in iterative structures. Vector data are used as input/output of the simulation and can be modified by the use of operator to compute distance, surface, geometry intersection, union and other spatial algorithm.

IV. MULTI-SCALE ONLINE DATA ANALYSIS AND VISUALIZATION

We apply our approach in a purpose related to data stream mining which is the process of extracting knowledge structures from continuous, rapid data records. This approach provides new exploration capabilities and strategies to represent, analyse and abstract a large amount of alternative simulation experiments by proposing an accurate representation of the model, abstracting data from the reference model and providing new way to interact with the model. We represent here the interdependence between various interacting entities. Such an online mining tool has already been applied in different application like modular interaction patterns identified in transcriptional co-expression networks [17], we then propose a generic solution relevant to any kind of agent-based model.

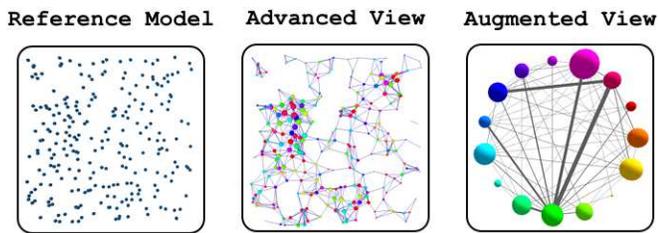


Fig. 3. In the advanced view, a proxy agent targets the reference agent (in the reference model) and an edge agent represents spatial interactions between two nodes. In the augmented view, a macro node represents the aggregated micro nodes of the same class and a macro edge represents the total number of links between two different kinds of node in the advanced view model.

A. Represent: Advanced Visualization

In order to ease the representation, we introduce the notion of proxy agent that will be used only for visualization purpose. Proxy agent can be seen as a delegate pattern, that insures the non alteration of the reference model, where the mirrored agent delegate visualization task to the proxy agent. 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. In Figure 3 the advanced view is a richer representation of the reference model. Proxy node represent the node of the reference model with a given color representing the value of a given parameter of the reference model and a size representing its degree in the graph. Edge entities represent spatial interaction between nodes.

B. Abstract: Augmented Simulation

From an already existing graph at a micro level (advanced view), a macro graph that clusters groups of agents and summarizes the number of links between each class of agents is created online as shown in Figure 3. A macro node represents all the aggregated micro agents of a given population. A macro node is represented by a sphere (of the same color of the aggregated agent) where radius evolves according to the number of aggregated nodes. 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. Different linkage functions can be used such as single linkage, complete linkage, average linkage or average group linkage.

C. Interact

This approach not only provides new tools to represent, extract and abstract inner dynamics but can be use to reinject the result of the online analysis in the reference model and interact with him by the use of controller so that the modeller has an immediate feedback thanks to the augmented view. From the macro graph representation a controller is linked to each macro node of the graph. The user can modify the number of agent that are represented by the macro graph in the reference model as shown in Figure 4 .

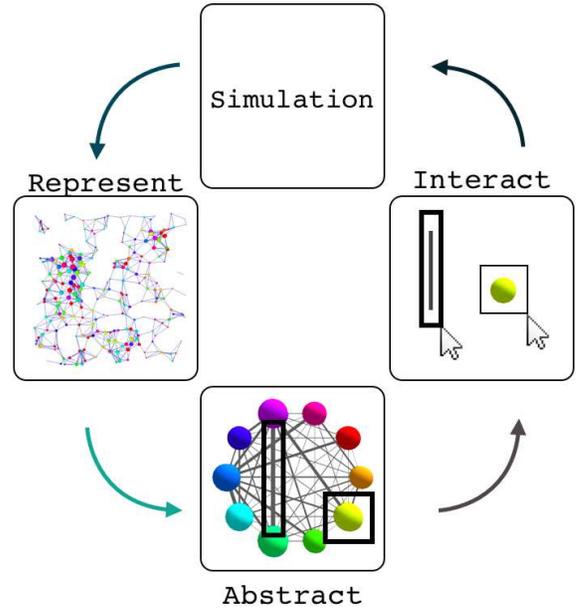


Fig. 4. Represent, abstract, interact.

V. EXPERIMENTS - RESULTS

After describing a general use of our approach in the online visualization and analysis of generic network, we show here a demonstration of the usability of our approach applied on a well known SIR compartmental model in epidemiology used to study many infectious diseases. The SIR model represent a population of human distinguished in three components: S for Susceptible, I for Infectious and R for recovered. We use color to represent the status of the human, size to represent its class of age and shape to represent its gender. The advanced view represent the spatial interaction network. The augmented view is a representation of the model where macro node represent the total number of S (Susceptible), I(Infected) and R (Recovered) people and the average number of link between them. The results presented in Figure 5 have been implemented in the Gama platform version 1.6.

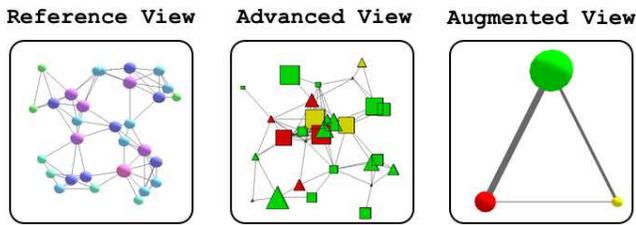


Fig. 5. Reference, advanced and augmented view of a compartmental model in epidemiology.

VI. DISCUSSION

The aim of this work is not to resolve in advance all the problems of representation possible that will emerge from the modellers, but to propose sufficient and generic concepts so that they can represent a simple and intuitive solution they imagined. Another approach that could be an alternative of the MVC design to target modellers or user model in general could be a reflection based on the construction of models of indicators that might be use as a complementary approach. The indicator approach would be more a final goal or objective whereas the MVC approach is more a tool. In term of performance, this design has been successfully implemented in GAMA but we need to study the cost in term of performance of the creation of such a framework. If the proxy agent is a copy of the mirrored agent it can be greedy to use this pattern as the number of objects working concurrently increases (this is a commonly known drawback of the MVC pattern). This is particularly true when the dataset is large. In this case, user interactions that go through a MVC abstraction could become non-realtime. Finally, in this approach even if feedback is possible at runtime by the use of controller, the model representation is described once and before the simulation and it could gain more interest to be able to dynamically change the representation of the model at runtime by using a dynamic language so that the user have a more agile way to interact with the visualization as proposed in agile visualization framework [18]

VII. CONCLUSION

We have shown here a design that separates the representation of a model from its execution to ensure that the visualization and the execution of a simulation are two independent processes. This approach, at the boundary between visualization and analysis, can be used to describe and represent indicator to both assess and explore a model using the same ABM paradigm used in the reference model. We have seen how view and controller may be seen as ABM model on which entities can be manipulated as an agent so that it can make emerge abstract datas that were not defined in the reference model. Finally we have applied this approach to the online analysis of ABM simulations (and applied it on a compartmental model in epidemiology) by representing interaction between entities in a view, then abstracting them at a macro level and using controller on macro node to alter the reference model. We

hope, that this approach for visualizing and interacting with ABM simulation will open new perspective for modellers to represent, abstract and interact with their model.

REFERENCES

- [1] B. Edmonds and S. Moss, "From kiss to kids—an anti-simplistic modelling approach," *Multi-Agent and Multi-Agent-Based Simulation*, pp. 130–144, 2005.
- [2] R. Allan, *Survey of agent based modelling and simulation tools*. Science & Technology Facilities Council, 2010.
- [3] Gama, "http://gama-platform.googlecode.com," 2013.
- [4] A. Drogoul, E. Amouroux, P. Caillou, B. Gaudou, A. Grignard, N. Marilleau, P. Taillandier, M. Vavasseur, D.-A. Vo, and J.-D. Zucker, "Gama: multi-level and complex environment for agent-based models and simulations," *AAMAS '13*, pp. 1361–1362, International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [5] M. J. North, N. T. Collier, J. Ozik, E. R. Tataru, C. M. Macal, M. Bragen, and P. Sydelko, "Complex adaptive systems modeling with repast simphony," *Complex Adaptive Systems Modeling*, vol. 1, no. 1, p. 3, 2013.
- [6] S. Tisue and U. Wilensky, "Netlogo: A simple environment for modeling complexity," in *International Conference on Complex Systems*, pp. 16–21, 2004.
- [7] A. Begel and E. Klopfer, "Starlogo tng: An introduction to game development," *Journal of E-Learning*, 2007.
- [8] AnyLogic, "http://www.anylogic.com/," 2013.
- [9] A. Crooks, A. Hudson-Smith, and A. Patel, "Advances and Techniques for Building 3D Agent-Based Models for Urban Systems," no. 001, pp. 49–65, 2011.
- [10] S. F. Railsback, S. L. Lytinen, and S. K. Jackson, "Agent-based Simulation Platforms: Review and Development Recommendations," *Simulation*, pp. 609–623, 2006.
- [11] A. Grignard, A. Drogoul, and J.-D. Zucker, "Online analysis and visualization of agent based models," in *Computational Science and Its Applications—ICCSA 2013*, pp. 662–672, Springer, 2013.
- [12] U. W. Daniel Kornhauser and W. Rand, "Design Guidelines for Agent Based Model Visualization," *Journal of Artificial Societies and Social Simulation vol. 12, no. 2 1*, vol. 12, no. 2, 2009.
- [13] R. Lamarche-perrin, Y. Demazeau, and J.-m. Vincent, "How to Build the Best Macroscopic Description of your Multi-agent System ?," pp. 1–18, 2013.
- [14] P. Taillandier, D.-A. Vo, E. Amouroux, and A. Drogoul, "Gama: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control," in *Principles and Practice of Multi-Agent Systems*, pp. 242–258, Springer, 2012.
- [15] A. Drogoul, E. Amouroux, P. Caillou, B. Gaudou, A. Grignard, N. Marilleau, P. Taillandier, M. Vavasseur, D.-A. Vo, and J.-D. Zucker, "Gama: A spatially explicit, multi-level, agent-based modeling and simulation platform," in *Advances on Practical Applications of Agents and Multi-Agent Systems*, pp. 271–274, Springer, 2013.
- [16] Processing, "www.processing.org," 2013.
- [17] E. Prifti, J.-D. Zucker, K. Clement, and C. Henegar, "FunNet: an integrative tool for exploring transcriptional interactions," *Bioinformatics (Oxford, England)*, vol. 24, no. 22, pp. 2636–2638, 2008.
- [18] M. Meyer, T. Girba, and M. Lungu, "Mondrian: an agile information visualization framework," in *Proceedings of the 2006 ACM symposium on Software visualization*, pp. 135–144, ACM, 2006.