COMPARISON OF THREE AGENT-BASED PLATFORMS ON THE BASIS OF A SIMPLE EPIDEMIOLOGICAL MODEL (WIP)

Kishoj Bajracharya, Raphael Duboz
Asian Institute of Technology
Pathum Thani, Thailand
kishoj@gmail.com, Raphael.Duboz@ait.ac.th

Keywords: Agent-based platform, Multi-agent system, SIR Model, Disease spread modeling, Verification

Abstract

With the extensive use of agent-based modeling and simulation, there are many agent-based platforms available. The objective of this paper is to compare 3 agent-based platforms: NetLogo, Repast, and Cormas based on the results of the simulation obtained from the same set of experimental scenarios. For this purpose, agent-based SIR model is chosen to study the pattern of the spread of an infectious disease within certain population over time. The methodology for the comparison is to design and implement the same agent-based SIR model in all these platforms, perform numerous experiments with the model and compare the outputs using cross-correlation analysis and similarity measure using Manhattan distance. The experimental results show that the agent-based platforms may inherently be biased by design and implementation choices, and they may lead to experimental results that can often be invalidated by other platforms.

1. INTRODUCTION

In recent years, 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. It enables to build agent-based model of complex systems as a collection of individuals, simulate the model and record the overall behavior of system with respect to individuals. With the rise of agent-based modeling (ABM), various agent-based platforms were developed. It is necessary to verify the correctness of the simulators developed in different platforms. Our aim is to check whether all these platforms give the similar results or different results from the simulation of ABM. We choose the three popular platforms NetLogo, Repast, and Cormas for verifying the correctness of the simulation of ABM. We wanted to compare these platforms and select the one platform which is reliable. In order to test the correctness of the results from these platforms, we need to compare results between three platforms. For the comparisons, we used the agent-based SIR model which is very simple to implement, easy to understand and adaptable to the real cases. To verify the correctness of 3 platforms, we should control the scheduling of actions of agent-based SIR model in these platforms. Simulations are performed under the same model and the same set of experimental parameters and then comparisons of the results are done by using statistical analysis. This paper is structured as follows. First, we begin with the related works in Section 2. In Section 3, we introduce the agent-based platforms in short. This is followed by methodology for the comparison of agent-based platforms in Section 4. Section 5 deals with the results of simulations and comparison of the agent-based platforms. Finally, we conclude the paper with conclusion in Section 6 and recommendation for future implementation in Section 7.

2. RELATED WORKS

Tobias and Hofmann [2] compared four agent-based tools: RePast, Swarm, Quicksilver, and VSEit. They rank the platforms by assigning the scores to represent the quality of the criteria of interest. The results of the experiments were to select platform based on the higher value of the weighted total score. They obtained the result in which four platforms are arranged in the decreasing order of their weighted total score were Repast, Swarm, Quicksilver, and VSEit. Railsback et al. [3] evaluated 4 main agent-based platforms: NetLogo, Mason, Repast, and Swarm based on the implementation of 16 versions of “stupidModel”. They compared platforms with one another and evaluated their capabilities through various metrics. The comparison of agent-based platforms was based on the outcome of the implementation of a series of models in each package but not by results of the simulations of the same ABM on different platforms. They suggested improving the platform Repast in its organization and design, to prepare complete documentation for the platforms except NetLogo, platform should provide statistical analysis tools and the need of research to understand the working of the simulation in the platforms. Their results showed that MASON was evaluated as the fastest among four platforms. Nikolai and Madey [4] 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. They assumed that the modelers will choose the toolkit based on their interest. They built a matrix that showed all of the characteristics of interest across one platform to see how a particular platform measures up as a whole for each of the characteristics. The literature provided by Laclavik et al. [5] showed a survey of agent based simulation platforms: NetLogo and MASON by implementing an exemplary scenario in the context of human behavior modeling. They evaluated the platforms considering 12 features as evaluation criteria. The evaluation criteria were loading and representing the environment and the scenario, creating and representing agents, behavior implementation, movement implementation, visualization, parameterization, model check-pointing, analytical tools, logging, performance, standards, and development environment. In the result, they noted that both platforms were almost equal in many features. They considered NetLogo was better in physical movement support and some analytical tools. However, MASON was best in its speed for faster simulation, strong separation of visualization and behavior models, better support for 3D environment, easier to integrate with other systems. Recently, there was a paper on press by Lytinen and Railsback [6] which will be appear in the proceedings of the fourth international symposium on agent-based modeling and simulation. In this literature, the authors reviewed and evaluated two latest agent-based simulation platforms: version 5.0 of NetLogo and the ReLogo component of Repast.They implemented the “StupidModel” series of 16 pseudo-models in both platforms. They found that ReLogo was more challenging to use and a less productive but ReLogo can convert NetLogo codes into ReLogo. The authors mentioned that NetLogo has developed into powerful platform for scientific modeling. They suggested that a modeler needs to be familiar with Groovy, Repast’s complex organization and Eclipse IDE in order to use ReLogo. They mentioned that there was less documentation of ReLogo than NetLogo. They could not distinguish the kind of the model that can be implemented in ReLogo which cannot be implemented on NetLogo. They found out that NetLogo was 20 times faster than ReLogo.

3. AGENT-BASED PLATFORMS

Agent-based platform is a technical architecture that supports the development of agents and agent-based components, provides the environment in which agents can actively exist and operate to achieve their goals. Serenko and Detlor [7] suggested that the need of agent- based platforms comes from the fact that existing OOP development platforms and compilers do not support all facets of agent development such as they do not address the implementation of agent features, agent interaction rules, common knowledge base, and communication language. Macal and North [8] mentioned that traditional modeling tools are not applicable for complex system where there is need of analyzing and modeling of system in terms of their interdependencies. Agent-based platforms are the tools used for modeling and simulation of an agent-based system. NetLogo is very popular ABM tool for modeling complex systems and is developed for educational purpose [9]. It provides rich integrated modeling environment with very simple modeling language. It provides simple visualization tools. It is easier to implement the models in NetLogo with less effort than any other platforms for agent-based simulation [10]. It is free to download and is an open- source. Programming in NetLogo is not object-oriented. Repast is a free, open-source, agent-based modeling and simulation library that can be used as a software tool to write agent-based models. Repast, the abbreviations for REcursive Porous Agent Simulation Toolkit, is a freely available agent-based simulation toolkit designed for social science applications [11]. RePast was developed at the University of Chicagoâs Social Science Research Computing Lab. It is an open-source agent based tools based on the Java programming language. It provides a core collection of classes for the building and running of agent-based simulations and for the collection and display of data through tables, charts, and graphs. Repast is fully object-oriented and platform independent. Common-pool Resources and Multi-Agent Systems (Cormas) is a generic agent-based simulation platform [12] and [14], developed in 1997 based on the VisualWorks programming environment which allows the object-oriented programming language Smalltalk [13]. The purpose of Cormas is to simulate agent based models of social and biological targets.

4. METHODOLOGY

The proposed approach for the comparison of the ABM tools starts with the implementation of same agent-based SIR model on each platform followed by simulations of agent-based SIR model. The results of simulations are saved on the file. After that we calculate the time series data of mean and standard deviation of 30 experiments per scenario. We consider 12 scenarios for our experiments taking mean time series and 12 scenarios for the sum of mean and standard deviation of time series. Lastly, we compare the time series data obtained from agent-based platforms using cross-correlation analysis and Manhattan distance analysis. The inputs to this model are population size, number of infected agents, probability of infection, infectious period, and grid size. The output from the model will be time series of susceptible proportion, infected proportion and recovered proportion. Before simulation, initialize all the input parameters required for the agent-based SIR model. Setup population (collection of susceptible and infected agents) in random position in space. If number of infected agent is greater than population size then display the error message. Initialize parameters for each agent like locate agent into random position on space, set initial health-state, the days since agent got infected, and the color of agent in space to identify the agent. The actions scheduled for each time step for agent-based SIR model are given below:

1. Increment each day or tick.

2. Perform agent movement on space in random direction.

3. Perform agent's interaction. Check the health_state of neighborhood of each agent within their infection range. If health_state of neighborhood of susceptible agent is 'I' then set Infected_Days := 1, the color of an susceptible agent as "red", and health_state of susceptible agent as 'I'.

4. Update the health_state for each agent. If health_state is I then increment the value of Infected-Days. If health_state is I and Infected-Days ≥ Infectious-Period then set Infected_Days := 0, color of an infected agent as "green", and health_state of an infected agent as 'R'.

5. Get the outputs from the model i.e. total simulation days, susceptible proportion, infected proportion, and recovered proportion.

6. Plot the graph of population proportion Vs Time.

7. Save the outputs from the model to a output file for backups.

The cycle of scheduled action repeats until there are no infected agents. This cycle belongs to the scheduling part of ABS. We try to control the scheduling for all platforms. Steps 2, 3 and 4 are the main mechanisms for our model which is common for all the platforms but steps 1, 5, 6 and of above scheduling differs from one platform to another platform due to the different model structure for platforms. In order to verify the correctness of three platforms, we should control the scheduling of actions in these platforms. As in Figure 1, we

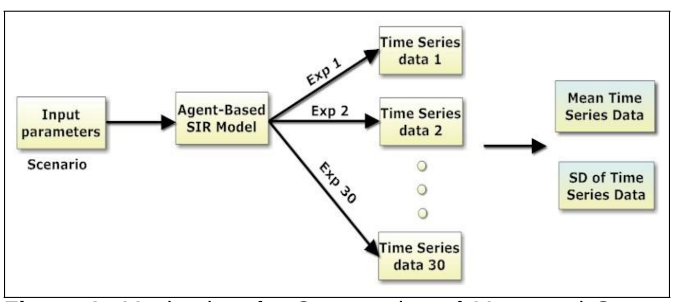

Figure 1. Mechanism for Computation of Mean and Standard Deviation of time series from simulation of Agent-Based SIR Model

perform 30 simulations per scenario to simulate our model for small grid size (21 × 21), medium grid size (101 × 101) and large grid size (301 × 301) for each platform. We consider a total of 24 scenarios (12 for mean time series and 12 for the sum of mean and standard deviation of time series) for performing simulations. For every experimental scenario, we perform the simulations with the constant population density of 20 agents per 100 cells (i.e. 0.2 agents per cell). Different agent-based platforms produce time series data in a different format. For comparing them, we need to change the format from different platforms to a common format for preprocessing step. After that we analyze the time series data obtained from all three agent-based platforms for 30 simulations. We compare the time series: susceptible proportion, infected proportion and recovered proportion using statistical analysis to determine whether all platforms have similar outputs for the same model and same initial conditions (parameters). We use two statistical tools: cross correlation test and Manhattan distance in order to observe whether time series data obtained from different platforms are similar. Depending upon the similarity of time series from the agent-based platforms, we determine the similarity between platforms.

Cross-Correlation is a measure of the similarity of shapes of the curves obtained from agent-based platforms. Let us consider two time series X_t and Y_t having 'n' number of simulation days. Cross-correlation coefficient function r_{XY} is defined as cross-covariance function as shown in equation 1

$$r_{XY}(\tau) = \frac{C_{XY}(\tau)}{\sqrt{C_{XX}(0).C_{YY}(0)}} \quad (1)$$

where, C_{XX} and C_{YY} are the sample variances of the time series data X_t and Y_t respectively and τ is the time lag. The value of cross-correlation coefficient function "r_{XY}" lies in between −1 to +1 i.e. $-1 \leq r_{XY} \leq 1$. The height of the cross-correlation coefficient function gives the measure of how correlated X_t is with Y_t and is independent of the magnitude of X or of Y. Greater the value of cross-correlation coefficient function between two time series data, greater will be the similarity between agent-based platforms. The cross-correlation operation is performed in between time series 1 and time series 2. We set a threshold for the cross-correlation coefficient function in order to decide whether two time series are similar or not. If the value of cross-correlation coefficient function exceeds the threshold then the shapes of the curves are similar otherwise, they are dissimilar.

Manhattan Distance (MD) is defined as the sum of the absolute differences for all the attributes of the two time series. It is useful in measuring the similarity between time series data because it can be used as metrics of dissimilarity. Let us consider following time series data are obtained from the simulations of SIR model on two agent-based platforms (1 and 2) for the same simulation time 'n'. Let the time series data obtained from platform 1 and platform 2 are denoted by T_1 and T_1 respectively then the mean MD is given by the following equation:

$$Dissim(T_1, T_2) = \frac{\sum_{i=1}^{i=n} |(a_i - b_i)|}{n} \tag{2}$$

where, a_i and b_i are the data from time series of platforms 1 and 2 respectively at i^{th} time. Let, $Sim(T_1, T_2)$ denotes the similarity between two time series data and is defined by following equation.

$$Sim(T_1, T_2) = \frac{1}{1 + \frac{\sum_{i=1}^{i=n} |(a_i - b_i)|}{n}} \tag{3}$$

The value of $Sim(T_1, T_2)$ lies in between 0.5 to 1. If the value is very close to 1 then the two time series data are very similar whereas if the value of $Sim(T_1, T_2)$ is very close to 0.5 then the two time series data are dissimilar. The average MD measures the similarity between two time series in terms of MD. Similarity tests are performed with time series of platforms 1 and 2 and compared the result with the threshold value. Depending upon the threshold, we decide the similarity between platforms.

Probability of Similarity (POS) between two agent-based platforms 'a' and 'b' is defined as the ratio of the frequency of the cases where agent-based platforms 'a' and 'b' are similar to the total number of experiments performed. Mathematically, POS is given as:

$$P_{Similar} = \frac{\#cases\,where\,platforms\,a\,and\,b\,are\,similar}{Total\,number\,of\,cases} \tag{4}$$

We calculate the POS between agent-based platforms for both cross-correlation test and Manhattan similarity test. Depending upon the POS, we will conclude the results.

5. RESULTS

The results of cross-correlation tests for different thresholds of cross-correlation coefficient function are shown in table 1. We calculated the POS for table 1 and the results are as shown in Figure 2. The POS between NetLogo and Repast remains the same for first three thresholds from 0.75 to 0.85, but if we further increased the value of threshold to 0.9 then the POS decreased. When the threshold value of 0.95 was chosen then the similarity decreased to 0.208. For the thresholds of 0.75 and 0.8, the POS between Repast and Cormas were same i.e. 0.083 but these platforms were almost dissimilar. The POS was zero for the threshold from 0.85 to 0.95 that means Repast and Cormas platforms are completely dissimilar. NetLogo and Cormas were highly correlated at the thresholds of 0.75 and 0.80 and the POS decreased slowly to 0.542 at threshold 0.90 and decreased further to 0.417 at threshold 0.95. The results of similarity tests using MD for different thresholds are shown in table 2. Using table 2, we

Table 1. Results of Cross-Correlation Tests for different Thresholds

Platforms	Count				
Threshold	0.75	0.80	0.85	0.90	0.95
NetLogo and Repast	12	12	12	10	5
Repast and Cormas	2	2	0	0	0
NetLogo and Cormas	23	20	13	13	10

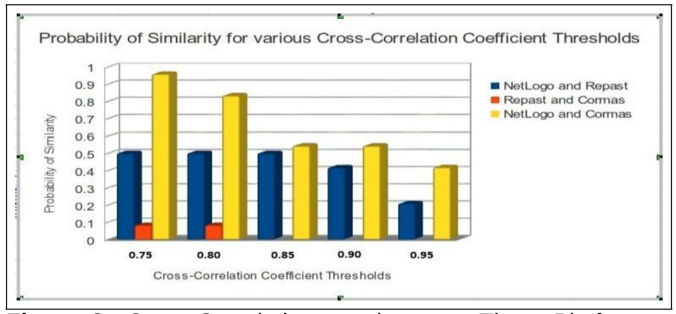

Figure 2. Cross-Correlation test between Three Platforms

calculated the POS between agent-based platforms which are shown in Figure 3. The POS between platforms NetLogo and Cormas was found to be 0.917 at threshold of 0.85 but as we increased threshold NetLogo and Cormas becomes dissimilar. We found out that the platforms Repast and Cormas were dissimilar for all thresholds from 0.85 to 1.0. NetLogo and Repast were very less similar at threshold of 0.85. However, as we increased the threshold value NetLogo and Repast becomes dissimilar.

Table 2. Results of MD tests for different Thresholds

Platforms	Count		
Threshold	0.85	0.90	0.95
NetLogo and Repast	7	5	1
Repast and Cormas	2	0	0
NetLogo and Cormas	22	8	0

From both the tests, we observed that POS between the platforms decreased with the increment in the thresholds. The platforms NetLogo and Cormas are similar at low threshold but if we increase threshold, the similarity decrease and becomes dissimilar. Repast and Cormas were more dissimilar than NetLogo and Repast. Hence, agent-based platforms do not give the similar results.

5.1. Discussion of the Simulation Result

Improper scheduling between different platforms affect in the simulation results but we tried to eliminate it by using the same scheduling for our model in all three platforms. The distribution of agent in the space and the movements of agents are stochastic and have important role in the dynamics of the

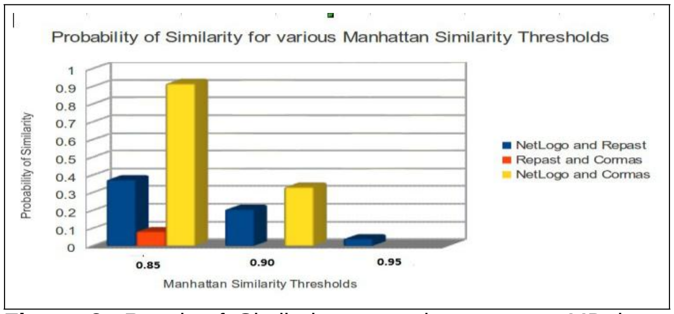

Figure 3. Result of Similarity test using average MD between 3 Platforms

complex system. For example: the agents move in the space in random directions to a grid which is not occupied by another agent. Cormas provides the built- in Smalltalk code to perform such a heuristic movement whereas in NetLogo and Repast, we checked whether the grid is occupied by another agent or not before moving an agent to the grid. The variation in the process of agent movements in different platforms affects in the similarity of the results obtained from these platforms. Agent-based platforms provide the facility for sorting a list of agents. According to [3], different platforms use different mechanism for shuffling the agent list. NetLogo does not describe the way scheduler orders the execution of the agent list whereas the agent list is shuffled in Repast using Repast class [3]. In Cormas, we used the built-in function for shuffling the agent list. Another fact for the variation in the results obtained from platforms is due to difference in the process of randomization of agent list in different platforms. Rails et al. [3] mentioned that model structure is one of the general issues during the comparison of platforms. Each platform has its own standard model structure. The model structure of Repast is less well-defined and it depends on the user whether to use separate model classes or to use single model class. NetLogo provides well-defined model structure that separates the processes of implementing and displaying a model. Cormas provides more well-defined structure for building the model from scratch to its simulation. For example: defining class for each entity, defining simulation organization, the observer (graphical tool), visualization (probes and spaces) and simulation. According to [3], each platform has a built-in method for scheduling repeated execution of actions but the mechanism of scheduling varies among platforms. If the platforms have different methods of scheduling in-spite of their same scheduling order of actions, the results obtained from them might vary. This means we cannot completely control the scheduling mechanism in the platforms like NetLogo and Cormas because we are not familiar with the model structure of them, but we have full control over the scheduling in Repast. If we cannot control the scheduling for these agent-based platforms then it is obvious that the results of these platforms vary. We tried to control only the scheduling of actions but there are some factors which we did not control. For example: number of methods, system calls, built-in functions, and methods for display. In NetLogo, we added modules for clearing and updating display and detecting neighbours in display. The behaviour of the model is found to be effected by the change in NetLogo versions because with the change of NetLogo versions, the agent-scheduling mechanism and random number generator may change. In Cormas, display update is automatic and we need to add the codes for termination of loop inside the scheduling. All these platforms have different mechanism for displaying the output on plot and display and saving the results in a file. Failure to control the scheduling of actions is one of the factors for the variations of the results between these agent-based platforms.

6. CONCLUSION

The key objectives in this paper was to compare agent-based platforms based on simulation results. We had implemented agent-based SIR model on 3 platforms: NetLogo, Repast, and Cormas, tried to control the scheduling and compared them based on the results of the simulation of SIR model. After performing simulations, we performed cross-correlation test to find the similarity in shape between the resulted curves from each agent-based platform. We observed the POS for different thresholds and found that the POS of two platforms depend upon the value of the thresholds. Higher the threshold value, lower is the POS between the platforms. We increased the threshold to 0.95, the value of POS decreased and agent-based platforms were dissimilar. From the result of the tests, it was found that NetLogo and Cormas were highly correlated (similar in shape) for the thresholds of 0.75 and 0.80 whereas Repast and Cormas were not correlated or dissimilar in shape for all thresholds. As we increased the value of threshold from 0.85 to 0.95, the platforms NetLogo and Cormas were dissimilar. The similarity tests using MD were performed where we had found that the platforms NetLogo and Cormas were similar at threshold of 0.80. However, the platforms NetLogo and Repast and Repast and Cormas were dissimilar for every threshold we checked. Agent-based platforms were found to be dissimilar with the increment in the thresholds. From these experiments, the agent-based platforms did not give the similar simulation results for the same model with the same set of experiments. From the results, it was found that three simulators gave different results for the same simulation so the results of simulation from these platforms cannot be used for making prediction or forecasting purpose because simulation results cannot be trusted. It is better to use agent-based platforms for the educational purpose to learn about the complex system. Moreover, we should consider agent-based platforms as the framework to develop the models in an easier way to help users to understand and develop the model of complex system in a new approach. Modelers use agent-based toolkits to perform experiments, to analyze the behavior of their system with their own way of thinking. Modifying the statement given by Page et al. [14] we can say that the main goal of the agent-based tools is not to make accurate predictions about the behavior of complex systems, but rather to provide a framework to help people develop new ways of thinking.

7. FUTURE IMPLEMENTATION

The one limitation of our research study is the use of only one agent-based model for similarity measure of agent-based platforms. All the results are based on the results of the implementation of agent-based SIR model on three platforms. As future work, we can use multiple agent-based models for similarity measure of agent-based platforms. It would be good research to know whether implementing other agent-based models also gives the same results or not. It would be interesting to check the similarity of two time series data for variable simulation days for complete simulation and to know whether there is similarity between agent-based platforms.

8. REFERENCES

[1] Nikolai, C., & Madey, G. (2009). Tools of the trade: A survey of various agent based modeling platforms.Journal of Artificial Societies and Social Simulation, 12(2), 2.

[2] Tobias, R., & Hofmann, C. (2004). Evaluation of free java-libraries for social-scientific agent based simulation. Journal of Artificial Societies and Social Simulation, 7(1), 2.

[3] Railsback, S. F., Lytinen, S. L., Jackson, S. K. (2006). Agent-based simulation platforms: Review and development recommendations. Simulation, 82(9), 609-623.

[4] Nikolai, C., & Madey, G. (2009). Tools of the trade: A survey of various agent based modeling platforms. Journal of Artificial Societies and Social Simulation, 12(2), 2.

[5] Laclavik, M., Dlugolinsk y Stefan, Seleng, M., Kvassay, M., Schneider, B., Bracker, H., et al. (2012). Agent-based simulation platform evaluation in the context of human behavior modeling. In Proceedings of the 10th international conference on advanced agent technology (pp. 396-410). Berlin, Heidelberg: Springer-Verlag.

[6] Lytinen, S. L., & Railsback, S. F. (2012, April). The evolution of agent-based simulation platforms: A review of netlogo 5.0 and relogo. In Proceedings of the fourth international symposium on agent-based modeling and simulation, at the 21st european meeting on cybernetics and systems research (emcsr 2012).Vienna, Austria.

[7] Serenko, A., & Detlor, B. (2002). Agent toolkits: A general overview of the market and an assessment of instructor satisfaction with utilizing toolkits in the classroom. Unpublished masterâs thesis, Michael G. DeGroote School of Business McMaster University.

[8] Macal, C. M., & North, M. J. (2006). Tutorial on agent-based modeling and simulation part 2: How to model with agents. InProceeding of the 2006 winter simulation conference (wsc)(pp. 7383). Monterey, CA: IEEE.

[9] Wilensky, U., & Stroup, W. (1999). Hubnet. http://ccl.northwestern.edu/netlogo/hubnet.html. center for connected learning and computer-based modeling, northwestern university. Evanston, IL.

[10] Wilensky, U. (1999) .Netlogo. http://ccl.northwestern.edu/netlogo/. center for connected learning and computer-based modeling, northwestern university.Evanston, IL.

[11] North, M., Collier, N., & Vos, J. (2006). Experiences creating three implementations of the repast agent modeling toolkit.ACM Transactions ron Modeling and Computer Simulation, 16, 1-25.

[12] Bousquet, F., Bakam, I., Proton, H., & Page, C. L. (1998). Cormas: Common-pool resources and multi-agent systems. InProceedings of the 11th international conference on industrial and engineering applications of artificial in telligence and expert systems: Tasks and methods in applied artificial intelligence(pp. 826-837). London, UK: Springer- Verlag.

[13] Page, C. L., Becu, N., Bommel, P., & Bousquet, F. (2012). Participatory agent-based simulation for renewable resource management: The role of the cormas simulation platform to nurture a community of practice.The Journal of ArtificialSocieties and Social Simulation.

[14] Page, C. L., Bousquet, F., Bakam, I., Bah, A., & Baron, C. (2000). Cormas : A multiagent simulation toolkit to model natural and social dynamics at multiple scales. In The ecology of scales. France.