International Journal of New Technology and Research (IJNTR)
ISSN: 2454-4116, Volume-2, Issue-10, October 2016 Pages 36-39

Agent-Based Simulation for Modeling Supply Chains: A Comparative Case Study

Keramydas Christos, Bechtsis Dimitrios, Aidonis Dimitrios

***Abstract* — Contemporary Supply Chain (SC) networks are complex and dynamic environments comprising of stakeholders and their fragile relationships, along with the associated interconnecting products, cash and information flows. The main objective of the SC is the creation of added value at every single node of the network. In terms of coordination, taking into account the dynamic nature of SCs, central coordination techniques are considered of high risk, as they are not resistant to changes and have a single point of failure. Decentralized management is a precondition for creating flexible and effective networks, as nodes need to dynamically move in and out of the SC and every node has a decision-making ability. Decentralized networks have no single point of failure, as every node can support the decision-making process according to the actual state of the network. Agent-based (AB) simulation platforms create decentralized networks of peer nodes, provide native mechanisms of communication and coordination techniques and are suitable for simulating the complex nature of SCs.**

In this paper, we first provide a literature review on AB simulation software used in supply chains modeling, we then describe an illustrative case study, and finally we develop two AB models using two distinct software platforms. The present work highlights the use of AB platforms agrifood supply chain modeling and can be used by academicians and practitioners as a practical example in order to incorporate AB techniques into the SC. The model-building details are provided for both platforms, as well as a critical analysis of the implemented models. The results demonstrate the incorporation of AB platforms to the SC ecosystem and compare the Agent-Based Modeling (ABM) capabilities of the alternative software platforms.

***Index Terms*—Agent-Based Simulation, Agent-Based Modeling, Agrifood Supply Chains**

I. Introduction

System modeling could potentially involve (i) process-based approaches that analyze the system's processes, (ii) analytical solutions with the use of complex mathematical models, and (iii) simulation techniques (Labarthe *et al*, 2007). Discrete event and Agent- Based Modeling (ABM) are considered among the most cited simulation techniques.

Supply Chains (SCs) are complex and dynamic networks that encounter significant changes over time. Agents, on the other hand, are autonomous entities that can act on behalf of real world actors and thus dynamically support the associated decision-making processes, taking into consideration both local and global knowledge about their environment. The nature of the SC promotes the use of agents as each SC stakeholder can be represented with an autonomous agent (Fig. 1).

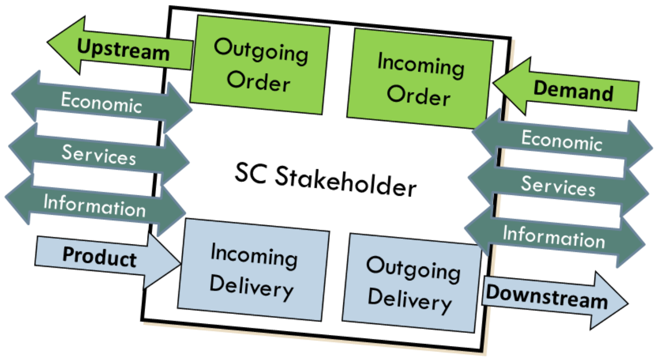

Figure 1. SC Stakeholders acting as autonomous agents

AB simulation has been identified as a useful tool to the development of certain decision-making processes in SCs mainly due to its somewhat "natural" correspondence between SC stakeholders and agents. The capability of representing the interactions between stakeholders over time, in a dynamic and distributed environment, is unique at the agent society. The AB approach allows for observing the behavior of each supply chain stakeholder over time, as well as of the SC as a whole. Multi-Agent System (MAS) platforms are composed of multiple agents that negotiate and cooperate in order to obtain their goals (O'Hare and Jennings, 1996). The MAS model could be implemented at (i) the Java Agent Development Framework (JADE) and (ii) the NetLogo Framework.

The remainder of the manuscript is organized as follows. In Section II we describe AB platforms that can be applied to the SC ecosystem. In Section III, we analyze the proposed model and we present two distinct AB implementations. Finally, in Section IV we wrap-up with conclusions and critical discussion.

II. Agent Based Platforms

A. *Agent-Based Modeling Frameworks*

The development of AB models is crucial for the proper and error-prone functionality of the final system. AB platforms allow for the rapid development of models with the use of inherent functions that implement the AB functionalities (behaviors, communication and coordination techniques, protocols, agent templates, etc.). The most cited

Christos Keramydas, Department of Mechanical Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece.

Dimitrios Bechtsis, Department of Automation Engineering, Alexander Technological Educational Institute of Thessaloniki, Thessaloniki, Greece.

Dimitrios Aidonis, Department of Logistics, Technological Educational Institute of Central Macedonia, Katerini, Greece.

AB platforms include:

- The JAVA Agent DEvelopment Framework (JADE, 2016) is a middle-ware that inherently complies with the Foundation for Intelligent Physical Agents (FIPA, 2016).
- NetLogo (NetLogo, 2016) is a multi-agent programmable modeling environment.
- AnyLogic is a multimethod simulation software (discrete event, agent-based and system dynamics simulation).
- StarLogo is an agent-based simulation tool from the MIT lab, which has a block programming environment and supports 3D animation
- RePast is an ABM suite which has a C++ and a JAVA based flavor.

For the model building the JADE and the NetLogo platforms are presented at the present paper.

B. JADE

JADE is an open source software framework which is implemented in Java programming language. Java ensures that the final application will be able to run on multiple devices and operating systems. JADE provides a middleware framework that complies with the Foundation for Intelligent Physical Agents (FIPA) specifications and has a set of graphical tools that support the model building and the management of the agents. JADE can help users with some development experience to build working agents with complex behaviors. Every JADE implementation has a distinct Agent Platform (AP) that can make several hosts interoperate. In an AP we can find the Main Container and we can build supplementary containers where agents can live and interoperate. A central registry contains all the agents of the platform and hosts three special agents that provide the basic JADE services; the Agent Management System (AMS) that provides unique names to agents and manages them, the Directory Facilitator (DF) that provides the Yellow Page Service of the platform and the Remote Management Agent (RMA) that implements the JADE Management Console itself. The ready to use suite of graphical user interface tools provides an easy way to manage the platform, the containers and the agents. The JADE Framework can be used as a tool for the development of MAS and has inherent FIPA support. Bellifemine (2007, 2010) gives an insight of the framework by analyzing both the basic and the advanced features in order to deploy a fault tolerant JADE platform.

C. NetLogo

NetLogo is a multi-agent cross platform programmable modeling environment for the simulation of agent-based models. The NetLogo's programming language is based on the Logo dialect and is extended to support agents. The graphical user interface is able to manage the code development and the agents' interactions and provides a friendly user interface for viewing the agents and controlling the model with the use of external parameters. Reporting of user defined and system variables is easily managed by diagrams and values at the user interface or external text files. NetLogo's basic agents are called turtles. They live and move over a grid of stationery agents called patches. Special link agents can be used to create connections among other agents. Links can be dynamically reconfigured in order to create ad-hoc networks.

Numerous extensions provide added value capabilities to the platform and provide functions that support the agents' behaviors. Extensions include clustering capabilities, statistical analysis tools, interoperation with third party software, development of goal-oriented agents that communicate using FIPA messages and many more. Agents interact and data flows can be easily collected from the Behavior Space tool which can coordinate multiple parallel runs of a model, using user defined parameters. The Behavior Space tool provides an easy way to run a sensitivity analysis on the model and make useful conclusions about its functionality.

III. MODEL DEVELOPMENT

The development of a model is the way to proceed from the real world problem to the simulation environment and it includes the following steps:

- Analyze real world examples and specific case studies.
- Select an abstraction level for the model and identify the involved entities, their properties and the underlined relationships.
- Map model entities to the AB scheme.
- Develop the agent based application to the specific software framework.
- Test the model's functionality under certain use cases
- Run the model in order to simulate the real world scenario.

A. Model Description

This paper examines an analytical model of a dual sourcing retailer (Vlachos *et al*, 2013). The grocery retailer is managing a single perishable product with a limited shelf-life (SL). The model examines ordering strategies for a single perishable product. The fresh product has a high price but after a given time frame the product loses its nutrition values and it becomes deteriorated. Two successive quality stages are considered: (i) the "freshness stage" during which the product maintains all its initial properties and is sold at a high price, and (ii) the "deteriorated stage" during which few of the above properties have decayed but the product is still marketable at a lower price and each one of them has the duration of the product's self life. The retailer has access to two supply channels; the distant and low cost regular supplier (RS) and the local but expensive emergency supplier (ES). The retailer has contracted agreements with two agricultural producers that can supply the required product quantities. The two suppliers differ in their distance from the retailer and in the price. Specifically, the RS is located offshore but is cheaper and rather inflexible (requires large ordering volumes). The ES is located closer to the retailer and is able to deliver any requested orders on short notice, but at a higher cost.

At the beginning of each stage the corresponding supplier can place a single order and the market demand follows an exponential distribution. The leftovers of the first stage are characterized as deteriorated products and are sold to the second stage while the leftovers of the second stage are considered of low importance. The emergency supplier may be used to increase the revenue and profit when the products of the regular order become deteriorated (during the second stage of their SL). The underlying costs include the classic inventory-related costs i.e. procurement cost (RS and ES) and lost sales cost while the only source of revenues are the unit price of each product sold to the customer (fresh and deteriorated products).

We focus on a MAS-based model for a perishable product with emergency replenishments that examines the above analytic model. The MAS takes also into account the freshness of the products. The agents of the system are the RS, the ES, the grocery retailer and the customer.

B. JADE Model Building

The JADE environment needs the use of an Integrated Development Environment (IDE). The most commonly used is the Eclipse platform (Eclipse, 2016) that is compatible with JADE. In order to develop the aforementioned model using the JADE framework we started with the agentification process. At this prototype we focused on the Retailer and the Customer Agents to monitor their behaviors and transactions while the RS and the ES are used as static agents (Bechtsis *et al*, 2015) (Fig. 2).

The Retailer Agent holds the business logic and the behaviors of the retailer. The Retailer Agent has certain behaviors that fulfill all the desired actions. Cyclic behaviors are used to simulate repeated actions: (i) organize the retailer's cycle when the store is ready to serve customer agents (the distinct stages of the cycle are implemented as a step behavior inside the cyclic behavior), (ii) organize the ordering policy of the retailer at the begging of each stage, (iii) update the retailer's inventory each time an order arrives, (iv) serve offer requests from customers attempting to buy products, and finally (v) serve purchase requests from customers that have agreed with the retailer to buy an offered product. The customer agent is primarily interested in fresh commodities and only if there is a lack of fresh commodities makes a second attempt in order to find deteriorated ones.

Two distinct classes represent the Retailer and the Customer agent. The software application can easily create numerous customer and retailer agents. The parameters of the program are variables inside the classes, while the model building logic is build on the main class. In order to give the initial state of the project we can either create separate user defined classes with a graphical user interface or build the business logic in the main class. The JADE platform provides auxiliary agents that support the monitoring of agent activities. A sensitivity analysis was conducted and the validity of the model was checked using the results from the analytical model (Vlachos *et al*, 2013).

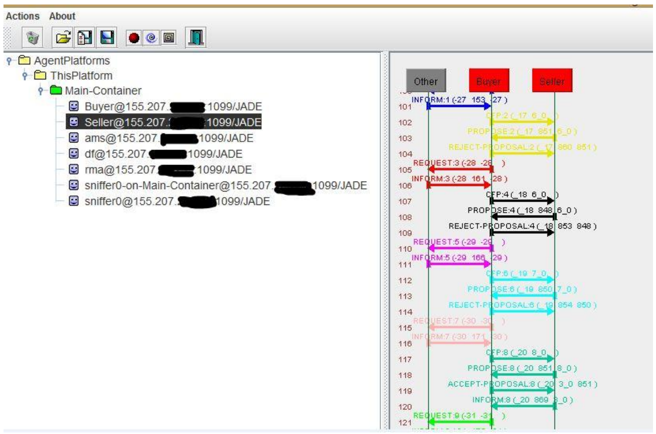

Figure 2. JADE GUI: Agent Interactions

C. NetLogo Model Building

The NetLogo interface is fully functional upon installation and has no need of an external IDE. The user must define the agents by means of the inherent NetLogo turtles, patches and links at the Code tab of the interface. No previous experience is needed for the set up of the NetLogo platform and many ready to use models exist in order to test the proper installation of the platform.

Four distinct breeds of agents were built in the agentification process, namely the Consumer, the Retailer, the RS and the ES. Moreover, special NetLogo agents named link agents were built to connect the agents and make them interoperate in a single network (Fig. 3). The business logic of the agents was built by the means of member functions. Functions were used to model the behaviors of the agents. The main program runs the functions by calling each one of them at a specific order following the distinct phases of the model.

It is worth to mention that the parameters of the model are easily controlled with user defined controls and can be changed both at the startup of the simulation and during the run time. The controls are easily embedded and handled from the user interface.

Finally, using the Behavior Space tool, it is feasible to conduct a sensitivity analysis by providing all the possible runs of the model for all the combinations of the user defined parameters. The step of every parameter can be adjusted and each combination can run multiple times.

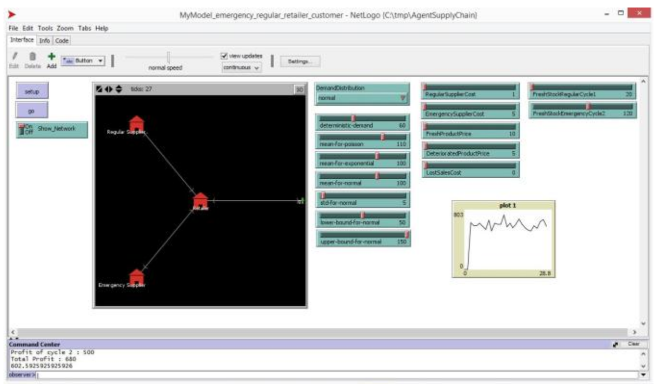

Figure 3. NetLogo GUI: Agent Interactions

IV. Conclusions and Critical Discussion

ABM techniques have reached an age of maturity and therefore can now provide added value to the digitalization of SCs in a "from cradle to grave" perspective. Intelligent agents can assist SC stakeholders in minimizing costs, increase flexibility, and avoid single points of failure while working on a 24/7 basis. In recent years, globalization has imposed major reconfiguration challenges for modern SCs, mainly in addressing mass customization requirements and coping with the ongoing economic crisis. The modern agrifood SC systems are complex, they have a dynamic character, and they also include a great number of entities, where each entity is trying to maximize its profit/values, while often is at the same time trying to maximize the overall SC's profit/values. MAS platforms are suitable for modeling and monitoring such kind of SC environments. In this context, the main priority/trend in modern SC networks is to immediately respond to customer demand, as well as to elaborately study customer profiles that can contribute back with critical insights on the management of modern SCs.

Our analysis clearly demonstrates that the incorporation of Agent Based Systems in SC management is a rapidly evolving research field, mainly due to the evident advantages of the decentralization approach. The present work is focused on two distinct AB platforms and could be further extended to address all AB approaches in the near future. Moreover, a further categorization of the agent-platforms will add extra value to the model-creation methods.

The models of our case study were developed at the most cited MAS platforms, namely JADE and NetLogo. The results indicate that both platforms can be conveniently employed in MAS simulation. In this context, the JADE platform appears to require a sufficient amount of programming experience, whereas the NetLogo platform requires a minimum of programming experience. JADE needs the JAVA programming environment in order to be able to implement the agents' classes and additionally the relevant work includes the use of a sophisticated IDE, whereas NetLogo makes use of the Logo language at the code development tab of the standard IDE. Both approaches have support for the FIPA protocol (NetLogo must use an external extension), they support the use of user-defined types of agent, local and global variables, and they also provide a graphical user interface for the visualization of agents.

Concluding, JADE appears to be a more sophisticated platform, providing greater flexibility and more robust models, but should be used by users who are willing to spend a great amount of time at the first steps in order to build more robust and expandable models. On the other hand, the NetLogo platform provides a simple user interface ready to be used by programmers with limited experience, focuses on rapid model development and provides a visualization environment with numerous ready to use extensions.

REFERENCES

[1] D. Bechtsis, M. Drakaki, D. Vlachos, "Using Agent-Based Modeling for the Design of Agrifood Supply Chains with Emergency Replenishments", in Proceedings of the 1st International Conference on Agrifood SCM & Green Logistics FP7 REGPOT Project: GREEN-AgriChains, 2015.

[2] F. L. Bellifemine, G. Caire and D. Greenwood, "Developing multi-agent systems with JADE", 2007, Volume 7, John Wiley & Sons.

[3] F. Bellifemine, G. Caire, T. Trucco and G. Rimassa, "JADE programmer quide", 2010 http://jade.tilab.com/doc/programmersguide.pdf

[4] Eclipse Platform, 2016. Available at: https://eclipse.org [Accessed 20 October 2016].

[5] FIPA, 2016. Foundation for Intelligent Physical Agents [online]. Geneva, Switzerland. Available at: http://www.fipa.org [Accessed 20 October 2016].

[6] JADE, 2016. Java Agent Development Framework version 4.4, [online]. Available at: http://jade.tilab.com [Accessed 20 October 2016].

[7] O. Labarthe, B. Espinasse, A. Ferrarini, and B. Montreuil, "Toward a methodological framework for agent-based modelling and simulation of supply chain in a mass customization context", in Simulation Modelling Practice and Theory, 2007, Volume 15, No. 2: pp.113-136.

[8] NetLogo, 2016. NetLogo Simulation Environment version 5.3.1, [online]. Available at: https://ccl.northwestern.edu/netlogo/index.shtml [Accessed 20 October 2016].

[9] G. M. O'Hare and N. Jennings, Foundations of distributed artificial intelligence, 1996, Volume 9, John Wiley & Sons.

[10] D. Vlachos, C. Keramydas, N. Tsolakis, E. Iakovou, "Design of Agrifood Supply Chains with Dual Sourcing and Emergency Replenishments". International Commission of Agricultural and Biological Engineers, Section V. CIOSTA XXXV Conference "From Effective to Intelligent Agriculture and Forestry", Billund, Denmark, 2013.

Christos Keramydas is a Senior Researcher at the Laboratory of Quantitative Analysis, Logistics and Supply Chain Management (LASCM), Department of Mechanical Engineering, Aristotle University of Thessaloniki (AUTh), Greece. He holds a Diploma degree in Mechanical Engineering (School of Engineering, AUTh), a MSc degree in Statistics and Modelling (School of Sciences, Department of Mathematics, AUTh), and a PhD in Logistics and Supply Chain Management (School of Engineering, AUTh). His research interests, and his teaching experience are spread in the fields of statistics, operations research, and simulation applications, as well as of supply chain management, logistics, risk management, and food security. He has participated in several national and international research projects, and he has published several papers in refereed scientific journals, and conference proceedings. Finally, he has been awarded several honors and scholarships during his career, thus far.

Dimitrios Bechtsis is a Lecturer at the Department of Automation Engineering, Alexander Technological Educational Institute (ATEI) of Thessaloniki, Greece. He holds a Diploma degree in Electrical and Computer Engineering (School of Engineering, Aristotle University of Thessaloniki (AUTh), Greece), and a MSc degree in Medical Informatics (School of Medicine and School of Sciences, AUTh). His research interests include information systems, supply chain management, and simulation techniques. He has participated in more than 20 research projects.

Dimitrios Aidonis is an Assistant Professor at the Department of Logistics of the Technological Educational Institute of Central Macedonia (TEI-CM), Greece. He received both his Diploma in Mechanical Engineering and his Ph.D. in Logistics and Supply Chain Management from the Department of Mechanical Engineering, School of Engineering, Aristotle University of Thessaloniki (AUTh), Greece, and his MBA from Staffordshire University, UK. His research interests focus on the areas of Applied Operations Research in Supply Chains, and Reverse and Green Logistics. He has participated thus far in several European projects. He has more than 50 publications in scientific journals and international conferences. He is a member of the Society of Logistics Engineers and the Council of Supply Chain Management Professionals.