Intelligent Agents and Multi-Agent Systems

TOWARD THE REFERENCE MODEL FOR AGENT-BASED SIMULATION OF EXTENDED ENTERPRISES

Jakieła Jacek, Litwin Paweł, Olech Marcin

Abstract: *The main aim of the research which partial results are presented in this paper is to develop full-blown reference model for creating simulation experiments supporting management of Extended Enterprises. Such framework can be used by modeler who would like to ask questions related to modeled business architecture in the context of managing part of the enterprise (e.g. supply chain) or the whole structure of Extended Enterprise. These questions may be answered with the use of computational analysis conducted with simulation model prepared according to proposed framework and simulation experiment process. The analysis may consider existing organization or the structure that is virtually designed and will be implemented after all characteristics are tuned. In order to facilitate the modeling process and reduce semantic gap between business architecture and model developed, agent orientation has been adopted as a modeling paradigm. Finally, the case study showing how the reference model may be used for analyzing the bullwhip effect in supply chain has been elaborated, and simulation experiment's results visualized and interpreted.*

Keywords: *Agent Based Modeling and Simulation, Extended Enterprise Modeling, Supply Chain Management, Agent-Oriented Computing.*

ACM Classification Keywords: *I. Computing Methodologies; I.2 ARTIFICIAL INTELLIGENCE; I.2.11 Distributed Artificial Intelligence; Multi-Agent Systems*

Introduction

Interaction and coordination costs have always been affecting the process of shaping business organizations structures and relationships they form with their partners. Development of Digital Economy and e-Business solutions has increased interaction abilities as well as reduced costs of coordination. All these phenomena have led to transformation of contemporary enterprises behavioral and structural characteristics and development of completely new and very complex business architectures. Now it is easier and cheaper to cooperate with business partners in industrial value chain than with departments inside organization [ITGI, 2005]. Therefore outsourcing became the commonplace and Extended Enterprises structures started to emerge. Managing of such enterprises as well as process of developing them requires really good understanding of the ways they operate and how the changes in their characteristics will affect the behavior of the structure as a whole. It is usually possible only with the use of organization's model and proper analysis method. In case of such dynamic structures analysis, simulation is often best suited approach. Due to the complexity of many Extended Enterprises, simulation is one of the few tools that can capture the dynamic nature of contemporary business structures in a realistic manner. The simulation process enables to conduct experiments with the model to understand the behavior of the system and evaluate strategies for its operation.

Simulation may be conducted according to several approaches which take into consideration different dimensions of simulation model e.g. evolution of the system over time – static vs. dynamic models, randomness element –

deterministic vs. stochastic models, how the state of the simulation changes – discrete vs. continuous models, how the system state is partitioned – system dynamics vs. multi-agent simulation [Paolucci et al., 2005]. Recently agent oriented approach to simulation – Multi-Agent Based Simulation (MABS) – is gaining wider and wider acceptance and its models and their advantages have been extensively described in many publications e.g. [North et al., 2007], [Weyns et al., 2009], [Yilmaz et al., 2009]. Contributions to the MABS domain are periodically published among others in Springer's LNAI series on Multi-Agent Simulation [Jakieła, Litwin, Olech, 2010], [Nuno et al., 2009]. Using agents in simulation models is based on the idea that it is possible to represent the behavior of active entities in the world in terms of the interactions of an assembly of agents with their own operational autonomy. The possibility to model complex situations whose overall structures emerge from interactions between individual entities and to cause structures on the macro level to emerge from models at the micro level is making agent paradigm a critical enabler in the modeling and simulation of complex adaptive systems [Yilmaz et al., 2009]. As will be shown later in the paper the Extended Enterprise is the problem space where agent oriented simulation is the best suited approach.

The process of simulation model development, when done from scratch, is usually related to quite big effort. It would be great if modeler will have the generic, ready to use structure and process that could be followed whenever new simulation model and experiment are elaborated. This is the place where reference model idea comes into play. The reference model may be treated as a comprehensive framework that facilitates the process of creating simulation model and experiment. The paper presents such a model for Agent-Oriented Simulation of Extended Enterprises. The presentation starts with the analysis of business architectures of contemporary enterprises and changes that are taking place with regard to their structures and behavior. Then the idea of reference model is explained as well as motivations why agent orientation is used. Finally the detailed description of proposed reference model follows.

Business Architectures of Contemporary Enterprises

Before the nuts and bolts of reference model will be presented, the organizational principles it is based on will be carefully analyzed and described. As there will be shown, the structure of reference model as well as modeling constructs are driven by several changes in contemporary organizations business architectures. Therefore the first issue the paper tackles is the concept of business architecture.

What is Business Architecture?

Enterprise architectures are commonly viewed as being mainly for ICT infrastructure; however today's organizations realized that the architecture in such a meaning hinders business rather than supports it. The main issue is to have the architecture that will eliminate semantic gap between business requirements and software support. The new approach focuses on unifying concept – the business architecture – that removes boundaries between business processes and ICT infrastructure [McCormack et al., 2003]. Business architecture dictates the shape of ICT support and is related to the knowledge that defines the business, the information necessary to operate business processes and ICT architecture needed to support organizational operations. It is rather difficult to clearly define the term business architecture, however we can use the following operational definition developed by Eriksson and Penker [Eriksson et al., 2000]:

Business architecture is an organized set of elements with clear relationships to one another, which together form a whole defined by its functionality. The building blocks of business architecture represent the structural and behavioral components of an enterprise system and show abstractions of the key processes and structures in the business. Good business architecture has the following characteristics:

- *captures the real business as truthfully and correctly as possible,*

- *focuses on key processes and structures of an organization at an appropriate level of abstraction,*
- *adapts easily to changes and extensions.*

Creating good business architecture requires taking into consideration the characteristics of contemporary enterprises. The next sections show the main shifts that have taken place recently with regard to enterprises structure and behavior. These trends have been called accordingly: From Hierarchy to Process Orientation and From Single Organization to Extended Enterprise.

From Hierarchy to Process Orientation

Functional orientation (see Fig. 1) means the focus on departments and tasks within departments. This functional mentality may be traced back to work of Adam Smith who first described the concept of breaking industrial work into simplest tasks [Smith, 1776].

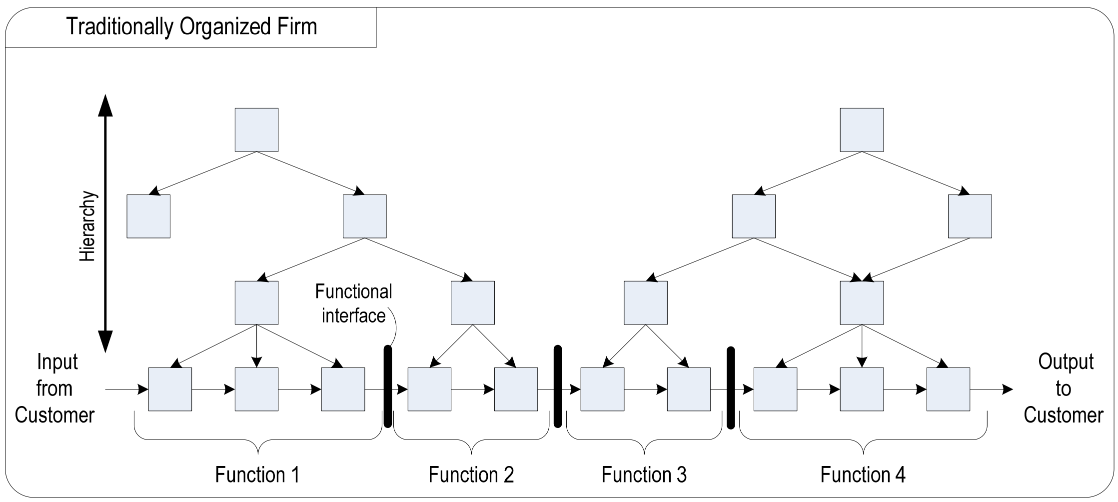

Figure 1. Traditionally organized firm [McCormack et al., 2003]

Organizations were following this model of business for almost 200 years. The main problem with functionally oriented firms is related to the lack of internal coordination of functions working together to satisfy customers.

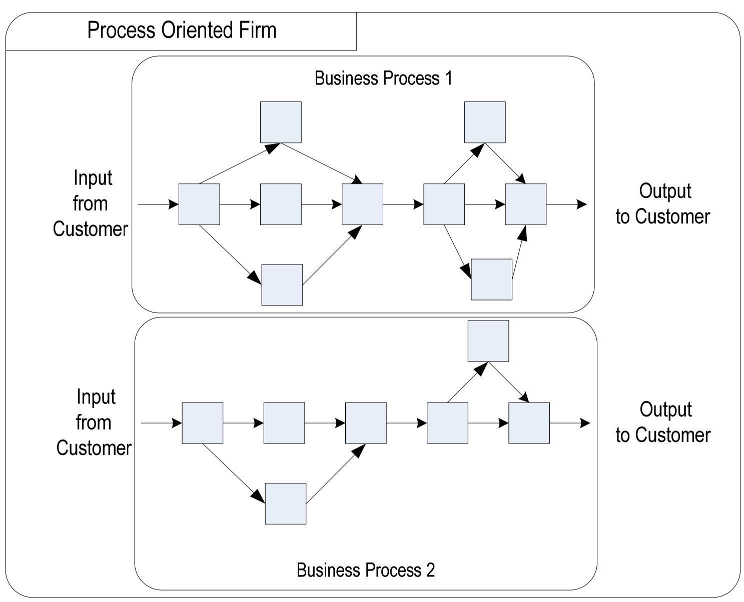

Figure 2. Structure of Process Oriented Firm

All the activities leading to performance improvement are linked to functional interfaces – the points where the control is passed from one function to another. These interfaces, if not properly designed, may be the source of poor quality, high costs and dissatisfied customers.

In order to improve functional interactions the business view should be reengineered. Organizational activities may be grouped in a manner describing them as a continuous process connected on one end with suppliers and on the other with customers (see Fig. 2).

The reengineering movement has been initiated by Michael Hammer in 1990 [Hammer, 1990]. According to his seminal work, the old, hierarchical organizational model is no longer relevant and something entirely different is needed. He developed the idea of process thinking, which is based on the process concept. Business process is defined as collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. The process thinking is cross-functional, outcome-oriented and essential to customer orientation.

Bearing in mind all the issues mentioned above it could be assumed that contemporary business architectures, in order to be efficient and effective, should be composed of strategic, customer focused processes that start with customers and emphasize the outcome. As Thomas Davenport suggested organizations must be viewed according to key business processes, not in terms of functions, divisions and products [Davenport, 1995].

Process orientation also affects the role of organizational actors and organizational structure of the process oriented firms. Process oriented firms are composed of autonomous, goal oriented organizational actors, who cooperate in business process teams to achieve the process goals. Organizational structures of process oriented firms usually take a form of adhocracy and are operating more on social principles than on mechanistic rules.

From Single Organization to Extended Enterprise

Contemporary enterprises have flexible, dynamic and more extensive boundaries than ever. Because of inherent complexity of today business, organizations must focus on whole processes, reaching out to business partners, suppliers and customers. Modern business architectures must be agile, otherwise they will not be able to cope with constant market changes. These changes are inextricably linked to the development of New Economy, which has three main characteristics: it's global, favors intangible assets (information, relationships, ideas) and is intensely interlinked. New Economy increases global competition what turns organizations to virtual integration enabling them focus on processes they can perform the best and someone outside the entity performs the rest. This phenomenon of extending an organization outside its traditional boundaries is commonly called Extended Enterprise. Extended Enterprise is a form of networked organization that is driven by standard enabling the network of participants to interact with significant interaction cost savings and high level of inter-firm connectedness. Extended Enterprise focuses on common goals and earns significantly greater value than its peers [IT GI, 2005].

To understand the evolution of such business architectures it may be useful to look at changes in interaction and coordination costs, which are playing a crucial role in shaping contemporary organizations, affecting the ways companies organize themselves and the relationships they build.

The standard business forms were based on the assumption that it was easier and less costly to interact with the departments of the firm than between outside entities.

ICT development pace has dramatically reduced interaction costs and increased interaction capabilities. It leaded to situation where it is easier and less costly to interact with business partners in industrial value chain than with business units inside the firm. Outsourcing became the commonplace and building and managing network of companies is a new competitive battleground. Because of all these changes Extended Enterprises are inherently decentralized and distributed [McCormack et al., 2003].

The Idea of Reference Model

Creating simulation model as well as planning an experiment from scratch is demanding and time consuming task; however it may be facilitated by providing modeler with the framework consisting of the generic structure of the simulation model, formalized behavior of its basic building blocks and the step by step process showing how to plan and conduct simulation experiment. The following sections present the reference model for agent-oriented simulation of Extended Enterprises.

What is Reference Model?

Unfortunately various papers and books describe very different things under reference model title. Reference model is often used to designate theoretical statements, technical architectures, or documentations of enterprise systems. The table 1 presents several definitions of the reference model concept.

Table 1. Definitions of Reference Model [Fettke et al., 2006]

Author	Definition
Bernus (1999)	Reference models capture characteristics common to many enterprises within or across one or more industrial sectors.
	Reference components provide normalized descriptions of key concepts of a given domain. They can be used a starting point for developing new applications similar to applications developed before in the domain thus extending reuse to the early development.
Fettke & Loos (2003)	Reference model represents a class of domains.
Frank (1999)	A generic reference model represents a class of domains.
Mertins, Bernus (1998)	An information system reference model is…a typical, or paradigmatic model, which describes the Information System or well identified part of it.
Mišic & Zhao (2000)	A reference model is a conceptual framework for describing system architecture, thus providing high-level specification for a class of systems.
Rosemann (2003)	Reference models are generic, conceptual models that formalize recommended practices for certain domain.
Schütte (1998)	A reference information model is a result of construction created by modeler who declares for IT and business people universal elements and relationships of a system as a recommendation with the help of a language in one point of time so that a point of reference is created.
vom Brocke (2003)	A reference model is an information model that people develop or use for supporting the construction of application models, though the relationship between the reference and application model can be characterized by the fact that object or content of the reference model is reused by the construction of the object or content of the application model.

Besides the problems with determining one generally accepted definition of reference model there are also other important aspects worth to mention. First issue is that different dimensions of business are addressed in the process of building reference models. The first dimension of the reference model is related to data models. Data models are considered reference models because they contain type entity classes describing organizational configurations. In general it is harder to find universal processes that are common to all businesses, and therefore process models are much rarer than data models.

Next issue is that the reference model, doesn't matter data or process, may be developed at different levels of abstraction. It usually depends on the scope that is covered by the model. If the whole industrial enterprise is modeled then only the high level of function or data breakdown is presented. More detailed description is possible if the modeler focuses on small problem domain e.g. reference model for inbound logistics.

Another aspect is related to descriptive or prescriptive character of the reference model, or to put it differently, is it the enterprise description enabling to better understand how it operates or it is the specification that can be used for system development or business process reengineering.

Finally from practical perspective, reference modeling can be used in different applications scenarios [Fettke et al., 2006]:

- Deriving a particular enterprise model – reference model can be adapted to the needs of particular enterprise.
- Validating enterprise specific models – reference model can be used as a benchmark for analyzing enterprise specific models and determine the gaps between them.
- Developing applications and simulations – reference model can be used as a framework for developing applications or computational representations of the enterprises created for simulation purposes. This is the scenario used in this paper.
- Selecting off-the-shelf package – reference model can be used as a tool for assessing different functionalities of off-the-shelf packages from the perspective of specific enterprise application.

For the purposes of this paper the following definition has been elaborated by merging several definitions from table 1.

Reference model captures characteristics common to many enterprises within or across one or more industrial sectors. It is also a conceptual framework for describing system (enterprise) architecture, thus providing high-level specification for a class of systems. Components of the reference model provide normalized descriptions of key concepts of a given domain. They can be used a starting point for developing new applications similar to applications developed before in the domain thus extending reuse to the early development. The reference model is supposed to answer the following questions:

1. *What are the main assumptions related to modeling scope?*
2. *What is a goal of developing the reference model?*
3. *How the conceptualization process is conducted and what are the basic modeling constructs the reference model has been based on?*
4. *How the structural and behavioral aspects are modeled?*

The above mentioned questions will be answered in the section of the paper describing the structure of developed reference model.

Because reference model presented regards agent based simulation, the next issues addressed are related to simulation as a method and agent approach to modeling and simulation.

Simulation as an Only Workbench for Extended Enterprises

Extended enterprises are very complex systems. This kind of business structures includes several components related to supplying, manufacturing, distribution, wholesaling and retailing. There are well documented reasons why simulation is only workbench for analysis and understanding behavior of Extended Enterprises. North and Macal [North et al., 2007] propose the following motivations:

- No one is able to understand how all parts of the system interact and add up to the whole.
- No one is able to imagine all the possibilities that the real system could exhibit.
- No one is able to foresee the full effects of events with limited mental models.
- No one is able to foresee novel events outside of their mental models.
- Decision makers want to get insights into key variables and their causes and effects.
- Decision makers want to make predictions of how system will behave. Thank to simulation they can get educated guesses and be provided with the range of possible futures.

Simulation can be used to address all these motivations. As will be shown in the next section, agent-based simulation provides new ways and insights not achievable with traditional simulation approaches.

Simulation Approaches

If we consider simulation of Extended Enterprises which can be treated as non-linear socio-economic complex systems, there are predominately two schools focusing on modeling, understanding and prediction of such systems behavior [Shieritz et al., 2003]. These methods are System Dynamics and Multiagent-based Simulation.

System Dynamics is a modeling method used mainly for the analysis of poorly structured problems, with a large number of interdependencies among the components of problem domain. This method has been derived from the cybernetic approach to systems analysis and allows one to describe systems in the form of interactive and combinational links. The method originally developed by Forrester is based on management theory, cybernetics and computer simulation [Forrester, 1961].

In System Dynamics approach, the real world processes are modeled as stocks, flows between stocks, and information indicating the size of flows [Forrester, 1968]. The main assumption is that internal cause and effect relationships determine the dynamic properties of the system. Most important impact on the functioning of the system has a structure within which decisions are made, rather than individual decisions and external turmoil [Shieritz et al., 2003]. Abstracting from single events and concentrating on policies instead, System Dynamics takes an aggregate view [Forrester, 1961]. Mathematically System Dynamics is a system of integral equations. The solution of such multi-equational system, describing nonlinear dependencies and use it to predict consequences of decisions under consideration, is usually possible only with an appropriate numerical method. System Dynamics gives reasonable tools to build models that provide information about the behavior of the system as a whole, enabling to improve the system development strategy.

Another widely used approach to analyze and understand business systems is Multiagent-based Simulation (MABS). Multiagent-based Simulation can be defined as the modeling and simulating of real world system or phenomena. In this approach the model consists of agents cooperating with one another in carrying out tasks and achieving goals. According to Davidsson [Davidsson, 2000] "[…] multiagent-based simulation should not be seen as a completely new and original simulation paradigm [...] it is influenced by and partially built upon some existing paradigms, such as parallel and distributed discrete event simulation, object-oriented simulation, as well as dynamic micro simulation […]". In fact, if the agents are considered as an extension of objects, MABS model can be seen as a consequence of defining the object-oriented system as a collection of autonomous entities, i.e. agents. The advent of Multiagent-based modeling has introduced an important innovation: behavior of complex systems with many active entities can be simulated by modeling individual entities and their interactions. Importantly, the operation of the system need not be defined a priori as set of equations, terms or logical statements, but the whole behavior emerges from individual objects behaviors, their interactions and impact of the environment. In MABS individual behaviors of agents adapt to environment and organize themselves.

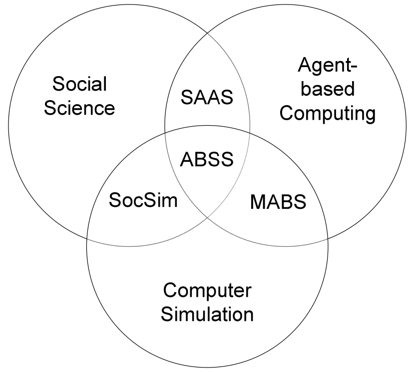

Figure 3. Map of the agent oriented research areas [Davidsson, 2002]

According to AgentLink Roadmap [Luck, 2003] "[…] Agent-based Simulation is characterized by the intersection of three scientific fields, namely, agent-based computing, the social sciences and computer simulation" (See Fig. 3).

Table 2. Characteristics of System Dynamics and Multiagent-based Simulation (MABS)

Feature	System Dynamics	Multiagent-Based Simulation
Model components	Stocks, flows between stocks with data indicating the size of flows.	Agents and their environment. With every agent the set of behaviors is connected. These behaviors enable agents to collectively achieve their goals.
Behavior	The behavior of a model is the result of structure. The structure is fixed and must be defined in advance. System Dynamics models are not capable of adaptation (structure modification with a goal to better perform in given environment [Holland, 1975])	System behavior is the result of interaction among agents. System components that are modeled as agents can communicate with one another, receive stimuli from the environment and exhibit a proactive behavior, which determines the way the system operates. Multiagent-based models can adapt by self-organization or with the use of genetic or evolutionary algorithms.
Modeling approach	Top-down approach is used. Aggregation is imposed by the modeler. The model represents the characteristics of the system as a whole. System Dynamics represents a group of macro modeling approaches.	Bottom-up approach is used. Multiagent-based modeling allows performing in-depth study of the macro level, which results from the actions undertaken by agents at the micro level.
Mathematical notation	System Dynamics Model is a system of coupled, non-linear first-order integral equations [Saleh, 2000]	There is no universally accepted mathematical formalism as a framework for Multiagent-based Simulation. Most researchers tend to logic-based formalisms [Inverno et al., 1997]

Agent-based computing is scientific area corresponding to computer science, in particular to modeling, designing and programming multiagent-based systems. Social science is the scientific field dealing with social entities interactions. It includes organization and management theory, social psychology and some areas of biology. Computer simulation is an area connecting the techniques of simulation with computer systems. According to Davidsson [Davidsson, 2002] the intersection of the areas mentioned above demarcates fields occupied by Agent-based Social Simulation (ABSS), Multiagent-based Simulation (MABS), Social Simulation (SocSim) and Social Aspects of Agent Systems (SAAS). Location of MABS at the intersection of agent-based computing and computer simulation can be accepted if we consider the application of MABS in simulation of manufacturing systems and processes. Due to the fact that vast majority of cases requires the inclusion of human factor in simulation of real world system or phenomena (e.g. manufacturing systems, Extended Enterprises, supply chains), it is assumed that social science also affects the MABS. Typical applications of MABS include simulations of business and social processes , as well as biological systems.

Selected differences between System Dynamics and Multiagent-based Simulation are presented in the table 2.

Approaches presented above represent important differences in the way the modeling process is conducted and possible applications are introduced. A major advantage of Multiagent-based Simulation is the ability to model emergency phenomena: behavior of a system as a whole is a result of the actions and interactions of autonomous, heterogeneous agents, while System Dynamics Model constitutes an indivisible whole. In the case of Extended Enterprise simulation, Multiagent-based approach makes it possible to better understand the business processes and their impact on the functioning of the enterprise as a whole. Very important feature of the simulation environment used for modeling today's enterprises is the ability to adapt to changing conditions without having to rebuild the model. Classical enterprise modeling tools, such as System Dynamics, do not take into account all the dimensions of modern organizations analysis (distributed nature of companies composing the Extended Enterprise, decisional autonomy, interaction dynamics). The advantages of Multi-agent Systems are a strong prerequisite for using this platform for developing and running the Extended Enterprise simulations.

Why Agent-Based Simulation?

This paper assumes that an agent paradigm is best suited for modeling and simulation of Extended Enterprises. This assumption was based on in-depth analysis of contemporary business architectures and insights provided by North and Macal [North et al., 2007], who answered the question: *When an Agent-Based Models are appropriate*? According to their hints using of agents should be considered in the following circumstances:

- Problem has a natural representation as consisting of interacting agents.
- Decisions and behaviors can be defined discretely, that is, with well defined boundaries.
- It is important that agents change their behavior and adapt.
- It is important that agents engage in dynamic strategic behavior.
- It is important that agents have dynamic relationships with other agents, and agent relationships form and dissolve.
- It is important that agents form organizations, and adaptation as well as learning are important at organizational level.
- The past may be a poor predictor of the future.
- Scaling up is important, and scaling up consists of adding more agents and agent interactions.
- Process structural change needs to be a result of the model, rather than an input to the model.

Besides motivations presented above, the following equally important have been added:

- Semantic proximity between contemporary structures and behavior of Extended Enterprises and multi-agent systems, what enables to reduce semantic gap between modeling constructs and organizational components. Multi-agent system is therefore very intuitive modeling metaphor that can be easily used in the process of building simulation model of Extended Enterprise.
- Better complexity management when agent paradigm techniques are used.

All these issues have been extensively discussed in [Jakieła, 2006], [Jakieła, Pomianek, 2009].

The Structure of the Reference Model for Agent-Based Simulation of Extended Enterprises

Main Assumptions and General Structure of Reference Model

Bearing in mind the results of analysis conducted the following assumptions have been formulated:

1. The model describes process–oriented organizations. It means that organizational activities are grouped in a business processes connected on one end with suppliers and on the other with customers.
2. Every process has a goal which is of value to customer.
3. Processes are conducted by autonomous organizational actors who interact in order to achieve the goal of the business process. Every actor has its own goals which are sub-goals of the business processes. In order to achieve the process goal, actors goals have to be achieved.
4. Organizations may form partnerships and integrate processes what leads to a structure of Extended Enterprise.
5. Reference model describes structural and behavioral aspects of Extended Enterprise.
6. The core of the reference model is a generic supply chain structure.
7. The model has hierarchical structure which is composed of three following levels:
    a. The highest level is the most general one and shows the basic components of supply chain as well as flows. These components, as will be shown later, have been mapped into multi-agent system architecture. Agents that constitute the system communicate according to flows patterns.
    b. The lower level covers the business logic of supply chain. This logic has been transformed into behavioral rules of the identified agents.
    c. The lowest and most detailed level describes the operations carried out by a producer.
8. The model does not take into consideration the following issues:
    a. Time elapsed between the moment when demand occurred and supplier takes note of it.
    b. Minimal quantity of an order – order is always fulfilled, doesn't matter how many items it contains.
    c. Minimal quantity of production order – production runs without any constraints related to economic quantity of items produced.
    d. Some operations and internal activities conducted by supply chain components such as: production planning, determining production capacity and demand forecasting.
9. The reference model is descriptive and prescriptive at the same time. It is descriptive because the analysis of its basic building blocks enables to better understand the business logic of Extended Enterprises. It may also be used as a specification for developing the agent-based simulation and therefore is prescriptive.

Problem Domain Conceptualization and Basic Modeling Constructs

According to Nilsson and Genesereth formalization of knowledge regarding specific problem domain begins with a conceptualization [Genesereth et al., 1987]. It includes the objects presumed or hypothesized to exist in the world and their interrelationships. The notion of object used here is quite broad and assumes that the objects can be concrete (merchandise sold) or abstract (order). Objects can be primitive (number of products in stock) or composite (agents). In general, an object can be anything about which it is important to say something. The set of objects about which knowledge is being expressed is called universe of discourse. Conceptualization defines also interrelationship between objects in the form of functions and relationships. Although many interrelationships may be defined usually only some are emphasized and others ignored.

The core of every Extended Enterprise is its supply chain. Therefore the reference model conceptualization has been based on the generic components of supply chain and their interrelationships (see Fig. 4).

Supply chain has several stages as is shown on the figure 4. It involves almost every aspect of production and provision of goods and services. Basic components are related to the stages and named accordingly supplier, manufacturer, retailer and consumer.

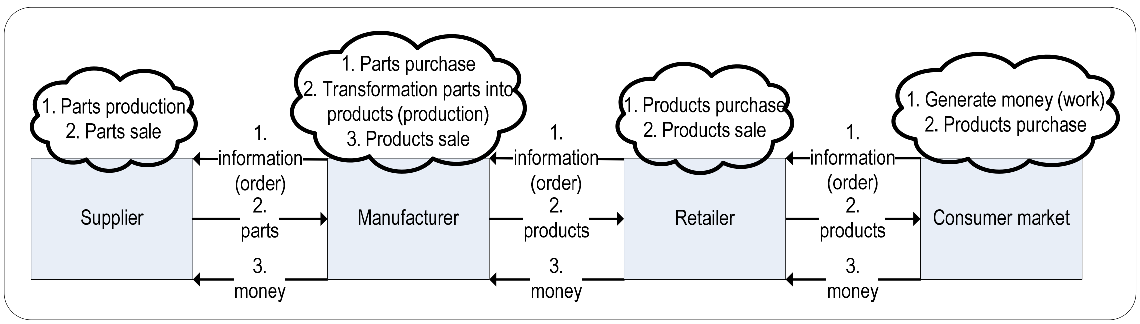

Figure 4. The Model of Supply Chain

There are also flows between basic stages. Two directions may be identified: downstream and upstream. Products flow downstream in response to orders that flow upstream as in case of payments. Real supply chains are dynamic business structures that form complex network between companies. The main problem is that such a network of interrelationships is the state of constant flux. That is the main reason why this reference model is simulation oriented.

After the conceptualization is defined, the next step is to select properly sensitive modeling constructs used in the process of developing the model. As it was mentioned before, different aspects of business can be taken into consideration in reference model. To some extent it determines modeling constructs used. For example if data aspect is considered modeler uses mainly entity type concept. When organizational processes are taken into consideration than functions are basic modeling constructs. Some more up-to-date reference models are using object concept as a basic modeling construct.

This paper proposes the approach that uses agents as a modeling metaphor. Next sections show the process of agents identification as well as agents' behavior definition.

Discovering Agents

The main features that make the system element candidate to be modeled as agent are the capability of the component to make independent decisions, the goal to focus the decision and the ability of other components to tag or individually identify the component [North et al., 2007].

When discovering the agents for the problem domain is worth to remember that each agent in agent based simulation has unique properties. Thank to this we can model the system as a set of interacting heterogeneous components. The uniqueness of each agent may be related to its behaviors, position it holds or resources it uses. Therefore building the model for agent-based simulation requires properly identifying the agents, defining the behaviors and taking into consideration the interaction patterns.

Discovery of agents has been based on the roles identification technique [Jakieła, 2006]. According to this approach it is assumed that every business organization is composed of the set of organizational positions. Every position is related to the set of business roles, that have goals organizational actors are responsible to achieve. In order to achieve specific goal, an actor has to have proper capabilities that may be based on the behavioral rules which are the topic of next section.

Based on the conceptualization presented in the previous section four agents have been discovered:

- Supplier Agent,
- Manufacturing Agent,
- Retailer Agent ,
- Market Agent.

Every agent has behavioral rules that drive its behavior during the simulation process.

Discovering Agents Behaviors

After all agents have been identified, the next challenge is to discover key abstractions for agents' behaviors.

Usually this stage is supported with knowledge engineering techniques for eliciting and organizing the knowledge of experts which is related to problem domain under consideration. These techniques are mainly based on structured interviews including focused questions domain experts are provided with. The results may contain some errors and biases but are often useful nonetheless. The behavioral rules presented in this paper are not derived as a part of knowledge engineering effort but have been drawn from the conceptual model developed by Vieira [Vieira et al., 2005].

It is important to remember that the knowledge engineering process should extend into fully specified model. This is possible thank to properly selected modeling methods. For this purpose, the UML activity diagram notation for specifying agents' behaviors has been selected. UML language is gaining widespread acceptance as a software modeling and design language and commonly used standard. It defines 12 types of diagrams (version 2.0) that can be used for visualizing different aspects of the system under consideration. As has been shown many times, combination of these diagrams may be used to fully document designs of agent based models as well as underlying knowledge [Padgham et al., 2004].

The diagrams presented on the figures show the business logic formalized in agents' behavioral rules and visualized with activity diagram notation.

<u>Supplier's behavioral rules</u>

Supplier Agent operates according to the doubling the order's quantity policy. This policy assumes that the demand in the next period will be equal of the demand in the current period.

If q denotes the number of items ordered, the quantity of delivery should equal $2q$. It is required to cover current producer's request and to have the final stock level enabling to cover future demand.

Figure 5. shows how the Supplier Agent operates. Firstly, it compares the doubled quantity of order received with the current stock level. Then, if it is greater than doubled quantity of order, the requested components are immediately sent to producer. Remaining number of items is greater than the number of items sent, and therefore supplier is prepared for the next order.

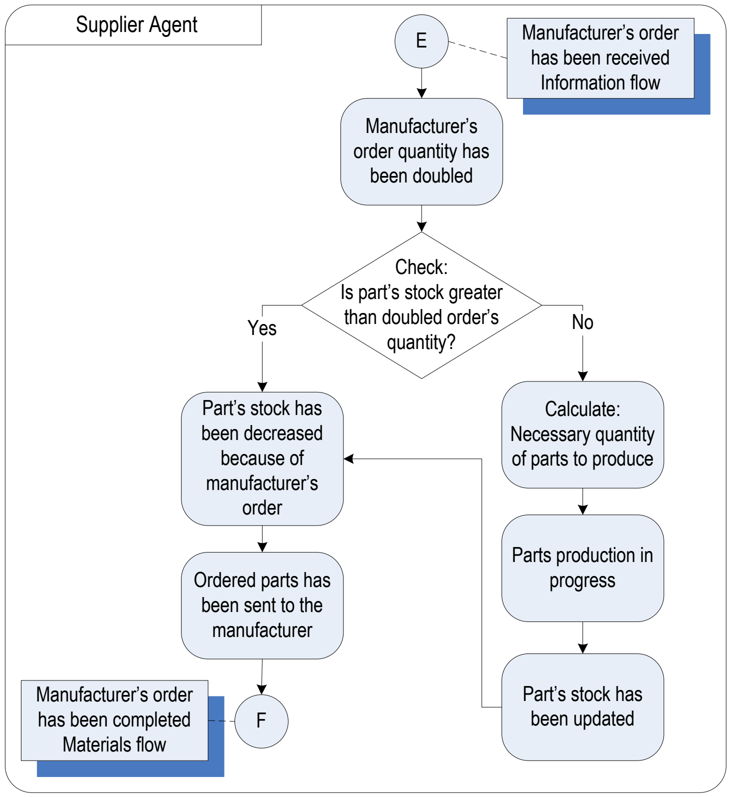

Figure 5. Behavioral rules for Supplier Agent

If the stock level is less than the doubled number of items ordered, supplier starts production of components (or orders items) right after current order is fulfilled.

Otherwise Supplier Agent has to start production process of components. Only after the required number of items is produced the producer's order may be fulfilled.

Producer's Behavioral Rules

An order received from Retailer Agent is an event that triggers proper behavioral rules of producer (See Fig. 6.).

In the first step Producer Agent doubles the number of items ordered to be prepared for the future demand. It guarantees that stock level will cover orders from the next period. As in case of Supplier Agent, Producer Agent compares the number of items in stock, and if it is greater than doubled number of items in current order the delivery is carried out, the stock level is decreased, and the whole process is finished. If the stock level is less than the doubled number of ordered items but enough to fulfill the order the items are sent to Retailer Agent. After that the Producer Agent calculates how many products are required to replenish the stock and fulfils the current order. Then production process starts and if there is lack of components (materials) needed, the order is sent to Supplier Agent.

Production process proceeds when all needed materials are received. It means that required number of components is taken from the stock and allocated to production. Final products are than used to replenish the products' stock level. The logic of production is depicted on figure 7.

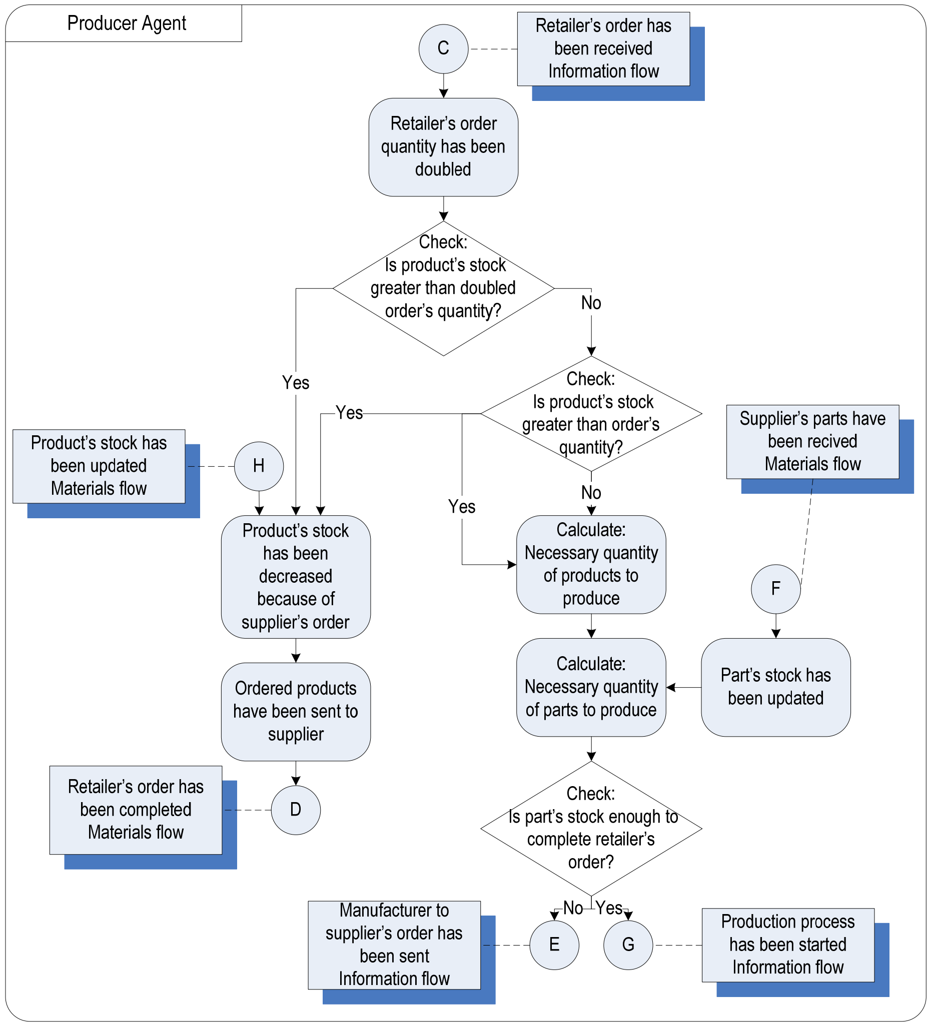

Figure 6. Producer Agent's Behavioral Rules

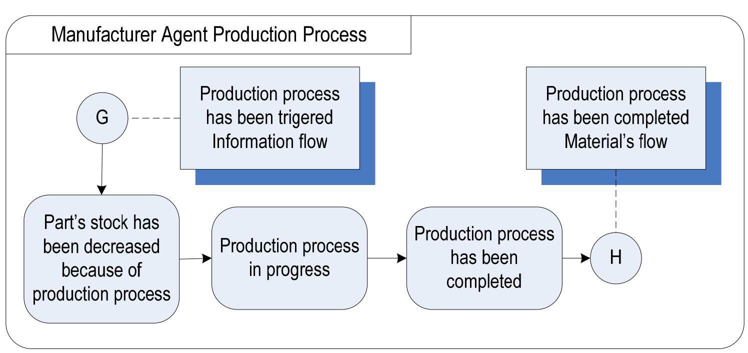

Figure 7. Procedure of Production Process

Retailer's Behavioral Rules

Structure and behavior of the Retailer Agent (see Fig. 8) is quite similar to the Producer Agent except for the business logic related to production and ordering processes. The starting point of Retailer Agent is the moment when an order is received, but in this case its source is the market. After the order is received and the number of order items is doubled, the retailer compares calculated value with its stock level. If the stock level is greater than the doubled number of order items, Market Agent is provided with products and process ends. In case the stock level is less than calculated value, but is enough to fulfill the order, the products are delivered and an order is placed to replenish the stock to the planned level. If the Retailer Agent does not have requested number of products it sends an order to Producer Agent and waits for delivery. When products are delivered, Retailer Agent fulfils the market demand and replenishes its stock level.

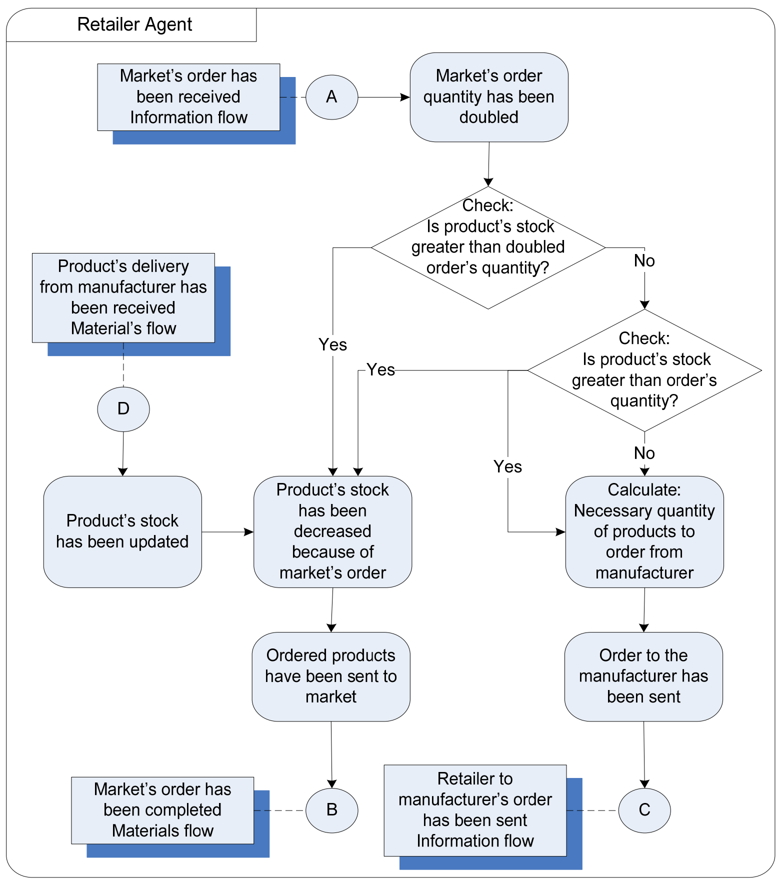

Figure 8. Retailer Agent's Behavioral Rules

Market Behavioral Rules

The Market Agent has only one goal – to generate the demand for products (see Fig. 9). This activity triggers the operation of the whole supply chain. The demand may be generated in three ways: according to demand forecasts, may be based on historical data or in the random manner with the specific density function of demand's distribution.

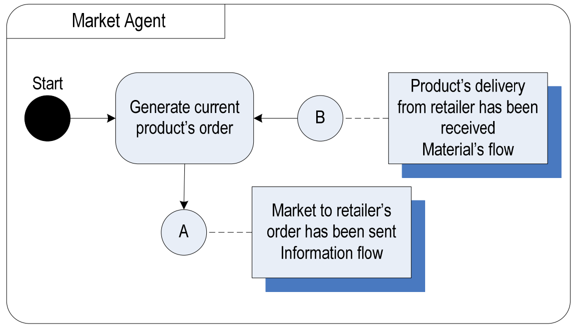

Figure 9. Market Agent Behavioral Rules

The Simulation Structure

The generic model of the simulation process is presented on the figure 10. The B connector comes from the environment preparation section (Warm-up Section) which is responsible for creating basic simulation model components (e.g. agents) and setting their initial parameters. It is specific to modeling and simulation environment selected. How it has been solved for the reference model under consideration is presented in the next section entitled Implementation of the Reference Model.

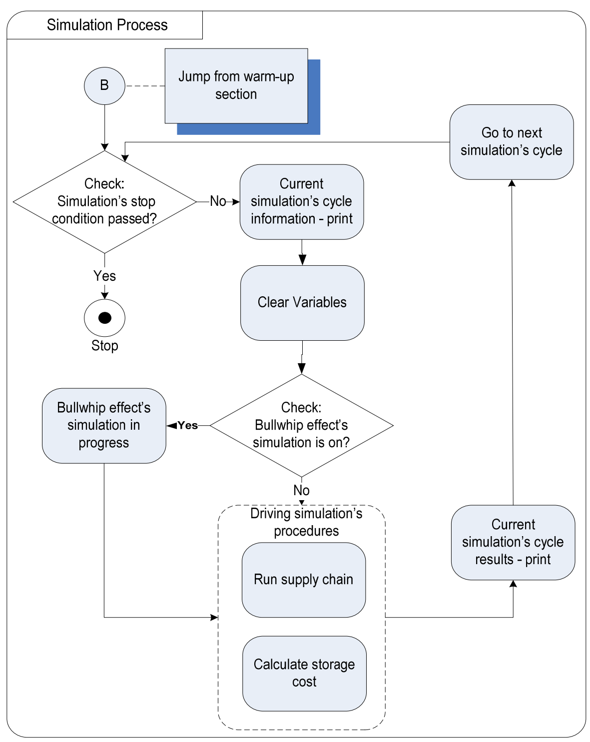

Figure 10. The generic model of simulation process

The first decision node is for checking if the stop condition has been met – if not, simulation proceeds. Simulation runs in the loops. Every loop constitutes the simulation cycle. Every simulation cycle has its own unique number assigned, which is automatically incremented by *Go to Next Simulation's Cycle* procedure. The number of simulation cycles may vary and lies in range from 1 to 1000. After next cycle starts, data related to it are printed on the system's console out. Just before Supply Chain procedures are run, variables values are set to initial states.

Next decision node checks if the cycle should include bullwhip effect phenomenon. After this decision is made, the procedures responsible for supply chain operation simulation are executed. The final procedure prints the results of current simulation cycle out. The printed data have the structure presented on the figure 11.

9 | 30 ; -26.7 ; 0 ; 0 | 4 ; 7 ; 92.36 ; 0 ; 3 | 3 ; 14.6 | 9.999985062E7 ; 0 ; Yes | 4.9

Figure 11. The row of data describing simulation cycle results

There are two data separators used in every row: "|" and ";". The former is put to separate the data related to supply chain components and the latter separates different values of the specific cycle's parameters. The table 3 presents interpretation of the values.

Table 3. Interpretation of simulation results

Values	Interpretation
9	Simulation cycle number
30	Number of parts the supplier has in stock
-26.7	Profit | Loss of Supplier
0	The number of parts, the supplier is supposed to deliver to producer in current simulation cycle.
0	The number of parts, supplier has to produce in order to fulfill producers demand as well as to replenish the stock level.
4	Producer Final Products' stock level
7	Producer Components' stock Level
92.36	Producer's Profit
0	Number of products, producer is supposed to manufacture in the current cycle (to cover the order and replenish the stock)
3	Number of products ordered by Retailer in current cycle
3	Final products' stock level at Retailer
14.6	Retailer's Profit
9999985062E7	The value of Market's Financial Resources
0	Market demand for current cycle
Yes	Flag showing if the products have been bought by the market in the current cycle
4.9	Time of the order fulfillment by producer (in hours)

The core of the simulation process is section responsible for Supply Chain operation (see Fig. 12). As is shown it consists of two main elements which are called Generate Market's Order and Fulfill the Order with Supply Chain.

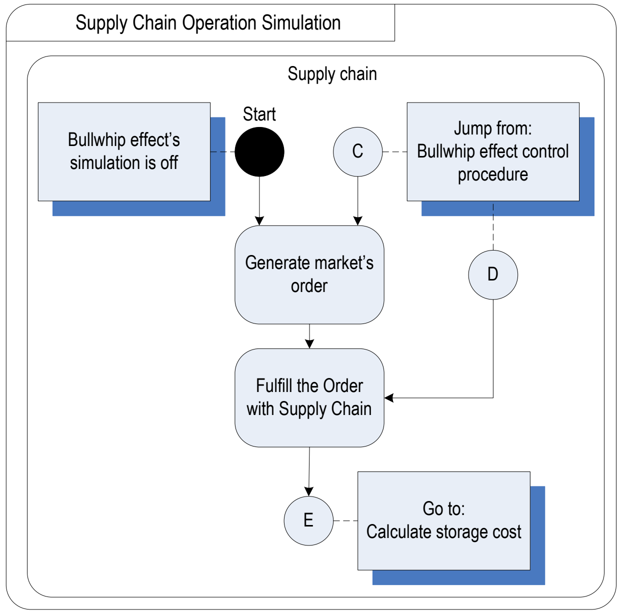

Figure 12. Supply Chain Operation Simulation procedure

The procedure entitled *Generate Market's Order* is depicted on the figure 13. It includes several decision nodes that check parameters such as demand level and market financial resources (money) level. If all conditions are met, the procedure *Try to place an order* is executed; otherwise the procedure called *Lack of an order in current cycle* is run.

After the order is generated by the market, its fulfillment is driven by the procedure *Fulfill the Order with Supply Chain*. The first step in this procedure is responsible for saving the fact that order has been placed. It supports the results' analysis because one can easily determine if the products have been delivered (or not) in the current cycle. Checking if Retailer Agent has enough level of stock is done according to behavioral rules presented in the previous section. Depending on the level of order's coverage, the following cases are possible:

1. Retailer Agent has proper level of stock to fulfill the current order and the number of products left equals the number of items currently ordered – *Fulfill the Order with Supply Chain* procedure is run.
2. Retailer Agent has enough products to fulfill the order, but the stock level left will be less than the number of items in current order – *Fulfill the Order with Supply Chain* and *Order Products* procedures are executed. Because the order has been fulfilled, the order status parameter is set to "closed" and therefore after ordered products are delivered they are not sent to the market. What is more, the variable Factor of Ordered Products is set to 1. It is used in the procedure of determining the number of products that have to be ordered.
3. Retailer doesn't have enough number of products to cover the current order – *Order Products* procedure is run, the order status is set to "open" and the order will be fulfilled after the delivery of producer will take place. The variable Factor of Ordered Products is set to 2, what means that number of products ordered has to be doubled. It is required to provide market with products demanded as well as to replenish the stock level.

The activity diagrams presented so far have been prepared on the high level of abstraction. It is obvious that in order to run the simulation several more detailed procedures were needed. Therefore every activity on the high-level diagrams has been further decomposed into next levels diagrams. The hierarchy of procedures required to running the Supply Chain Simulation is showed below.

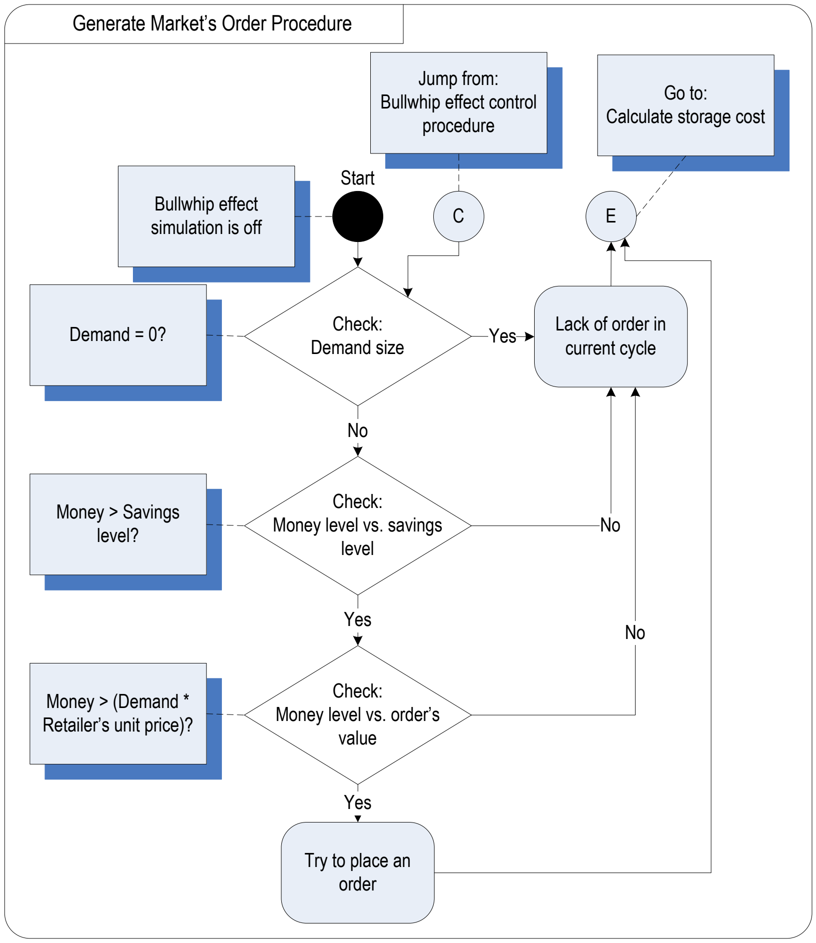

Figure 13. Generate Market's Order procedure

Run Supply Chain:

- Generate Market's Order
  - Lack of an Order in Current Cycle
  - Try to Place an Order
- Fulfill the Order with Supply Chain
  - Execute Market's Order
  - Order Products
    - Check Part's Stock Level at Supplier
    - Send Products to Retailer
      - Send Delivery to Retailer
      - Manufacturer's Production has been started
        - Manufacturer's Production process in progress
        - Order Parts from Supplier
          - Send Delivery to Manufacturer
          - Production at Supplier has been started

Indents show the hierarchy of procedures defined for simulation.

The Simulation Experiment Basic Steps

When using simulation, the experiment creation process is of great significance. The reference model presented in the paper assumes that experiment will be done according to the following steps.

1. **Identify and formulate a problem that is to be solved with simulation experiment.** It is important to have the list of questions articulated, which are supposed to be answered with the use of simulation runs.
2. **Create the model.** Definition of the real system model, which should provide a compromise between the detailed mapping of real system (impeding its implementation and validation), and simplification that suppresses the similarity between the model and real system, but making model easier to build. This step is extensively supported by reference model presented in the paper.
3. **Choose the metrics**. After the problem is formulated it is important to have set of indicators to monitor the state of the modeled system and to draw conclusions as well as to find hidden rules or laws.
4. **Set the initial values for simulation parameters**. The sample parameter is duration of simulation or number of simulation cycles. During every cycle all necessary tasks will be done, of course, only if their preconditions are met. The example of other parameters may be demand distribution (Normal, Poisson or Exponential distribution), parts and products prices or stock levels.
5. **Run the simulation**. After model is fully elaborated, and all variables are set up, the simulation may start and operate the predefined number of cycles during which values of variables are captured and saved.
6. **Calculate the metrics**. Metrics enable to better understand modeled object or process. Based on this data the simulation analysis is done.
7. **Interpret simulation results**. The last phase is related to the careful analysis of simulations results and calculated values of variables. It may lead to discovering new rules or dependencies that may be used in the real system management process.
8. **Validate the model**. It is based on a comparison of real system results and results of simulation, carried out with parameters corresponding to the actual operation of the system. The model can be considered valid (verified positive) if it (repetitively) reflects the behavior of real system.

Next sections present case study that shows how to use this scenario in Supply Chain analysis.

Implementation of the Reference Model

Selection of an Implementation Environment

Although one can implement model with a conventional programming language (most frequently Java; however other languages also can be used) this is usually hard way to start. Because simulation models involve similar building blocks (only small variations are present), commonly used elements have been put together in the form of the libraries, frameworks or full-blown environments. The best known are Swarm, Repast and Mason. They can be integrated with .Net Framework and programs written in C#, VB and Pyton as well as with Java environments. The main problem with these solutions is that they are quite complex and it can take months before one take an advantage of the full range of features they offer. What is more, the modeler has to use rather low level languages such as Java or C# what can be really daunting.

It is worth to mention that environments intended for other purposes can also be used for agent-based simulation. There are many examples of simulation done in Ms-Excel, statistics package R, MatLab or Mathematica. Nevertheless, the easiest and most effective way of conducting agent-based simulation is to use the package dedicated to this purpose [Gilbert, 2007].

Because main assumption of the reference model is to decrease the complexity associated with the modeling process, more suited are modeling environments that provide complete systems in which models can be created, executed and visualized. In this area the best known are StarLogo, AgentSheets and NetLogo.

The Reference Model proposed in the paper has been implemented in NetLogo, which is considered as the most popular simulation environment. It includes several facilities such as tools for building user interface or system dynamics modeler. The environment is free of charges for educational and research purposes. The comparison of modeling environments is shown in the table 4.

Implementation Process

The Multi-Agent simulations implemented in NetLogo IDE consist of three following stages:

1. Definitional part – includes definitions of agents' types ("breed"), global variables ("globals") and variables related to every agent (e.g. agent's name).
2. Preparatory part (to setup) – procedures setting up the environment for simulation.
3. Simulation definition part (to go) – procedures definition that drive the core of the simulation process.

All the parts and their temporal relationships are shown on figure 14.

Table 4. The comparison of agent-based simulation environments [Gilbert, 2007]

	Swarm	Repast	Mason	NetLogo
License	GPL	GPL	GPL	Free but not open source
Documentation	Patchy	Limited	Improving but limited	Good
User Base	Diminishing	Large	Increasing	Large
Modeling Language	Objective-C, Java	Java, Pyton	Java	NetLogo
Speed of Execution	Moderate	Fast	Fastest	Moderate
Support for GUI Development	Limited	Good	Good	Easy to create
Built-in ability to create movies and animations	No	Yes	Yes	Yes
Support for systematic experimentation	Some	Yes	Yes	Yes
Ease of Learning and Programming	Poor	Moderate	Moderate	Good
Ease of installation	Poor	Moderate	Moderate	Very easy
Link to GIS	No	Yes	Yes	No

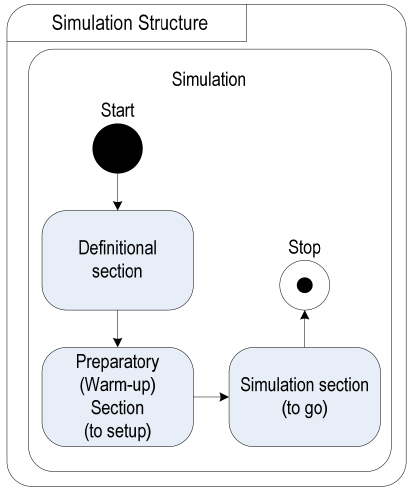

Figure 14. The Simulation Structure

<u>Definitional Part</u>

According to conceptualization described in previous section this part defines four types of agents: Market Agent, Retailer Agent, Manufacturer Agent and Supplier Agent (see Figure 15).

Every agent has the set of variables defined. All variables are presented below.

Supplier Agent has such variables as:

- components_level – number of components in stock,
- components_price – the component's unit price,
- components_cost – the component's production cost,
- profit – supplier's profit.

Manufacturer Agent's variables are the following:

- prod_stock_level – product's stock level,
- prod_price_level – product's unit price,
- components_level – component's stock level,
- components_price – component's unit price,
- to_deliver – number of products ordered by Retailer Agent in current cycle,
- to_produce – number of products to manufacture in current cycle,
- profit – producer's profit.

Retailer Agent's variables are:

- prod_stock_level – product's stock level,
- price – product's unit price,
- profit – retailer's profit.

Market Agent's variables are:

- money – market's financial resources,
- savings_level – the level of saving on the market,
- earnings – financial resources generated by market during one simulation cycle,
- demand – market's demand.

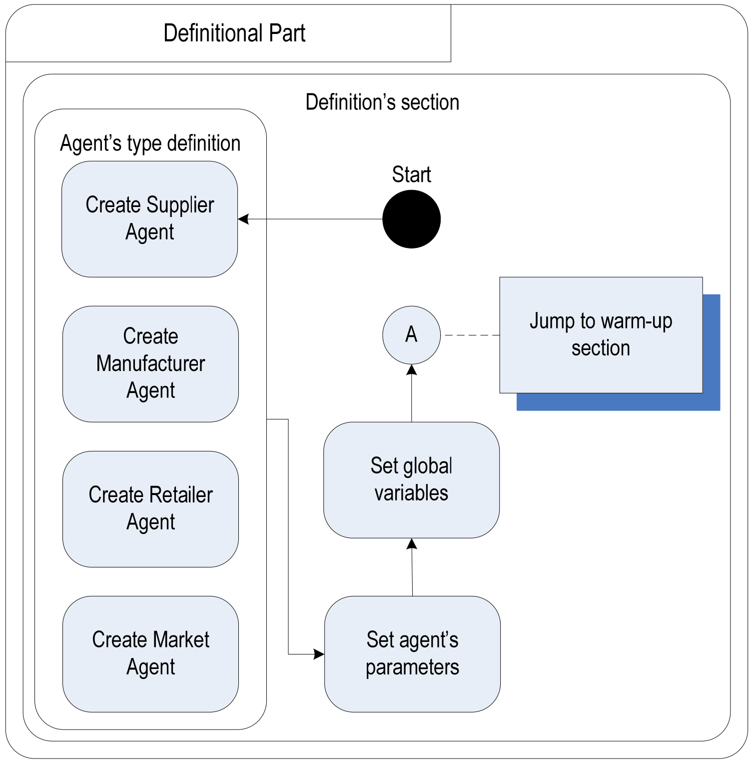

Figure 15. The structure of Definitional Part

Besides the agents' variables, this section includes also the definition of global variables that are available during the whole simulation process. Global variables defined for the simulation have been listed below.

- prod_price – unit price of final product at manufacturer (producer),
- comp_price – unit price of component at supplier,
- dmd – market's demand,
- production_quantity – number of products to be manufactured by producer,
- components_to_order – number of components ordered by producer,
- components_to_produce – number of components produced by supplier,
- factor – the quantity factor related to components ordered by retailer,
- production_factor – the quantity factor related to products manufactured by producer,
- components_factor – the quantity factor related to components produced by supplier,
- orders_state – order's state at retailer,
- components_delivery_state – order's state at supplier,
- prod_to_retailer_order_state – order's state at producer,
- text – variable used for saving simulation results,
- temp_text – auxiliary variable used for concatenation,
- purchase – flag used for showing that in the current cycle products were sold,
- time – used for measuring time needed for order fulfillment,
- bull – used for showing that bullwhip effect will take place in the current cycle.

All presented variables have been defined in simulation environment and used in agents behavioral rules during implementation stage.

<u>Preparatory Section</u>

This section is responsible for setting up the simulation environment. *Clear simulation* procedure is predefined in NetLogo environment and is used to clear the memory for storing new simulation results data.

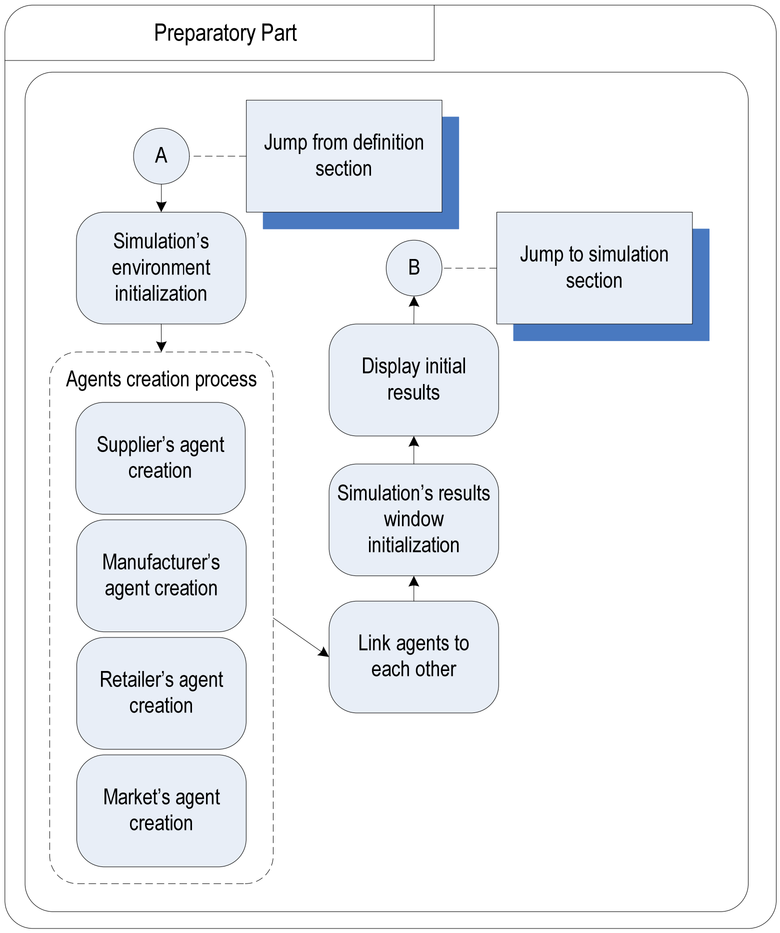

Figure 16. Structure of Preparatory Part

Procedures that instantiate agents are responsible for setting up initial values for variables defined in definitional section. Next procedure creates visual connections between communicating agents. Finally presentation layer procedures prepare the simulation window and write down simulation initial values as well as simulation results what helps in results analysis process.

The final step of this stage was implementation of simulation procedures according to agent behavioral rules and simulation structure described in the previous sections.

Described activities are depicted on the figure 16.

Case study – Bullwhip Effect in Supply Chain Analysis

The following paragraphs present short case study, which illustrates how the multi-agent-based simulation can be used as a tool supporting supply chain management process.

The simulation experiment has been elaborated in the steps presented below. All the steps have already been explained in the section entitled *The Simulation Experiment Basic Steps*. Description of each step reveals the design decisions which have been made with regard to Supply Chain modeling and simulation processes.

1. **Identify and formulate a problem that is to be solved with simulation experiment**. The main goal of the multi-agent based simulation is to check *how the bullwhip effect may influence the retailer's delivery time.*

2. **Create the model**. The application of multi-agent approach to the process of simulation model development has been based on the framework presented in the paper. The conceptualization has led to four main model components:
   - consumer market agent – this element is responsible for buying and consuming products,
   - retailer agent – its main goal is to buy products from manufacturer and to sell them to consumers,
   - manufacturer agent – it is obliged to purchase materials from supplier, transform them to final products and sell final products to retailer,
   - supplier agent – this component produces materials and sells them to manufacturer.

Very important part of the model is the logic of agents' operation, which is formalized with behavioral rules presented in the previous sections. During the runtime, agents communicate using the following flows:

- goods flow – in the direction from the supplier through manufacturer and retailer to consumers,
- orders (information) flow – from consumers through retailer and manufacturer to supplier.
- money flow – shows how the money circulates in supply chain.

3. **Choose the metrics**. The equation (1) describes total delivery time measured from the moment when the retailer has ordered products to the moment when products have arrived to retailer.

$$T_T = T_D + T_P + T_{DC} + T_{PC} \qquad (1)$$

where:

T_T – total delivery time, T_D – time of delivery between manufacturer and retailer, T_P– production time at manufacturer's floor (it's skipped if manufacturer has products' stock greater than the order lot), T_{DC} – time of delivery between supplier and manufacturer (included only if manufacturer product's stock is smaller than order lot, what is more, supplier has enough part's to fulfill manufacturer's order), T_{PC} – part's production time at supplier's floor (included only if supplier part's stock is not sufficient to fulfill manufacturer's order).

The algorithm realizing this function is implemented in main part of the simulation process. Metrics' values are calculated automatically at a runtime.

What is more, during simulation experiments other metrics have been calculated e.g. average time of different delivery types – it will be described in the step number 7.

4. **Set the initial values for simulation parameters**. Because the goal of the simulation experiment is to compare how the bullwhip effect may influence total delivery time of retailer, minimum two simulation runs have to be executed. The first run does not include the bullwhip effect and the second one takes it into consideration.

This case study contains two simulation experiments and two runs in each of them. Initial parameters of simulation runs are presented below.

- First experiment:
    - duration of the simulation: 300 cycles,
    - part's price: 2.50,
    - demand distribution: Normal,
    - mean of distribution: 6.0,
    - standard deviation of distribution: 1.5,
    - safety part's stock at supplier: 25 pcs.,

- safety part's stock at manufacturer: 20 pcs.,
- "bullwhip effect" simulation: first run - Off, second run - On,
- cycle when bullwhip effect occurs: first run – N/A, second run - 250,
- strength of the bullwhip effect: first run – N/A, second run - 20,

- Second experiment:
    - duration of the simulation: 500 cycles,
    - part's price: 2.00,
    - demand distribution: Poisson,
    - mean of distribution: 5.0,
    - standard deviation of distribution: N/A,
    - safety part's stock at supplier: 25 pcs.,
    - safety part's stock at manufacturer: 20 pcs.,
    - "bullwhip effect" simulation: first run - Off, second run - On,
    - cycle when bullwhip effect occurs: first run – N/A, second run - 450,
    - strength of the bullwhip effect: first run – N/A, second run - 20,

5. **Run the simulation.** The simulation has been run four times, with parameters values as presented in previous step. The results have been saved in text files and used in the process of metrics calculation.

6. **Calculate the metrics.** Collected data was re-formatted (precision, conversion of dots to commas) before the calculation process. After that, some statistical operations were done such as: search maximum, minimum and calculate average. The final results of the experiment will be presented and interpreted in the next section.

7 . **Interpret simulation results.** In this step the most interesting results will be visualized and presented.

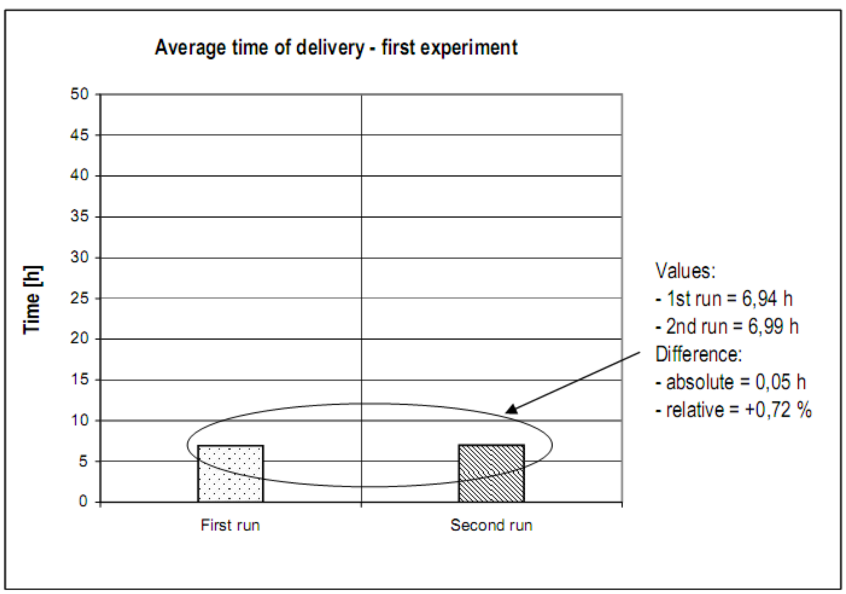

Figure 17. Average delivery's time in the first experiment

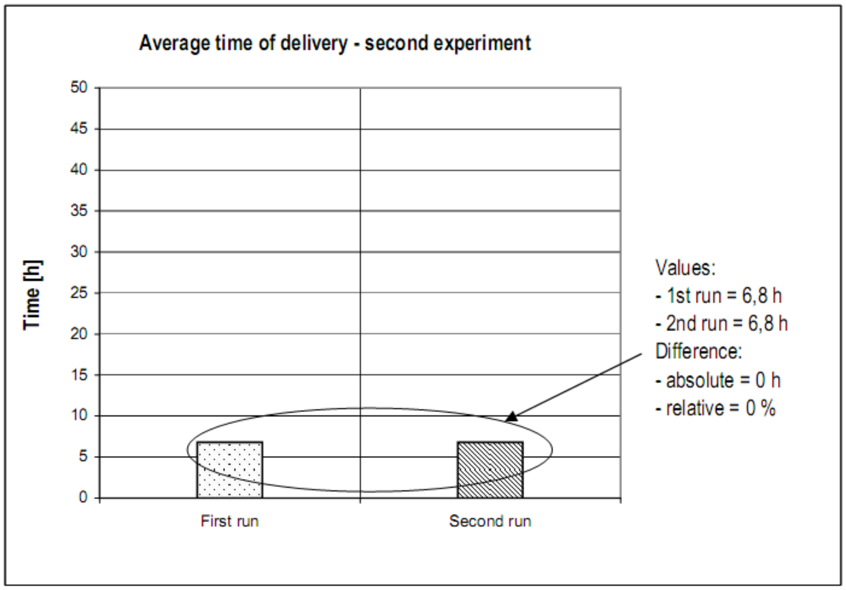

Figure 18. Average delivery's time in the second experiment

As one can see on the figures 17 and 18, an average time of deliveries in the first and second experiment was almost the same – around 7 hours.

What is more, turning on the bullwhip effect simulation did not have a spectacular influence on the results (compare first and second run on the graphs 17 and 18). Figures 19 and 20 show direct deliveries between manufacturer and retailer (A type delivery) – transported goods are final products. Average time of those deliveries was approximately 4 hours, and also here, the bullwhip effect did not have an important influence. One can notice strange situation on the figure 20. Average time of direct deliveries is shorter if the bullwhip effect simulation was enabled, and it is because of randomly generated pattern demand in the first and second experiment as well as in the first and second run. To make the whole analysis process more clear and understandable, static pattern demand should be considered.

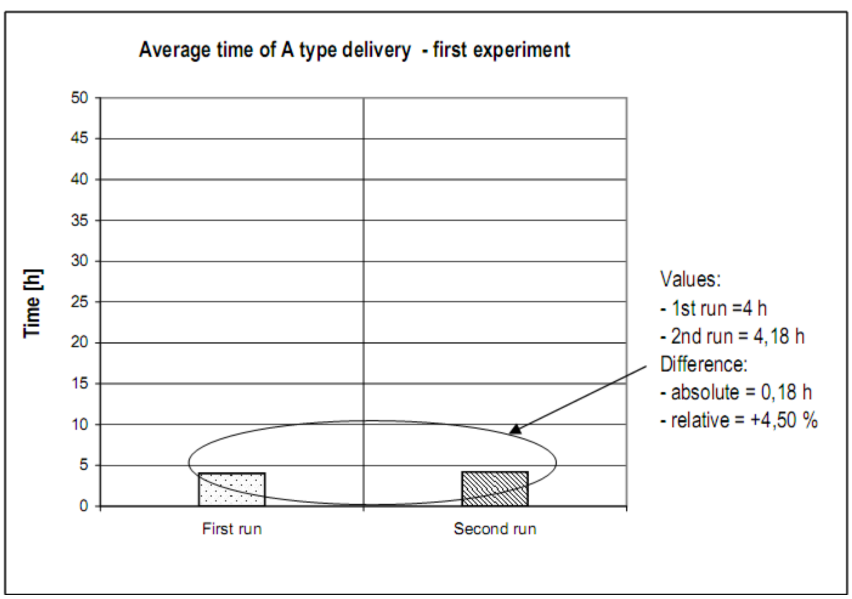

Figure 19. Average A type delivery's time in the first experiment

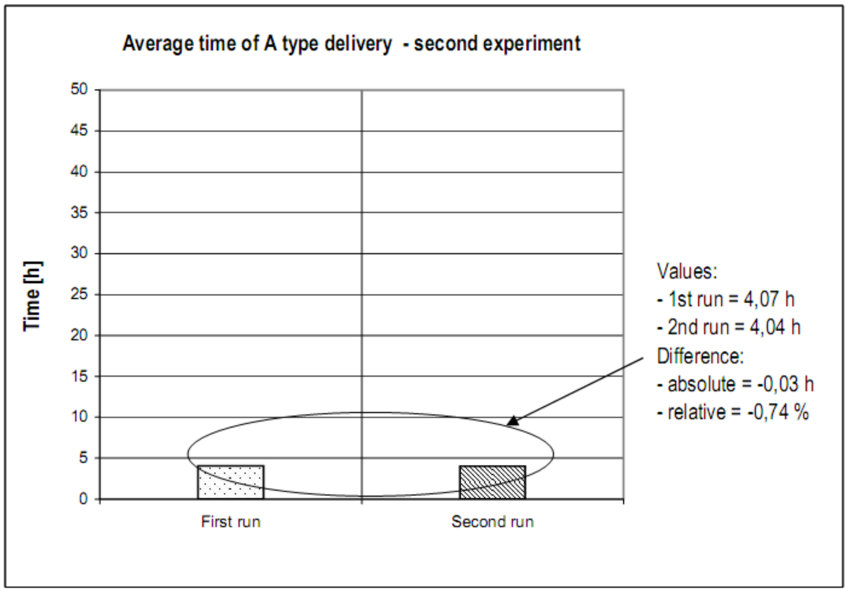

Figure 20. Average A type delivery's time in the second experiment

Next two graphs (Fig. 21 and Fig. 22) show average time of deliveries in B category. It is composed of three stages:

1. delivery of parts between supplier and manufacturer,
2. transformation of parts into products (manufacturer's production process),
3. delivery of products between manufacturer and retailer.

Average delivery time in this case was equal around 9 hours.

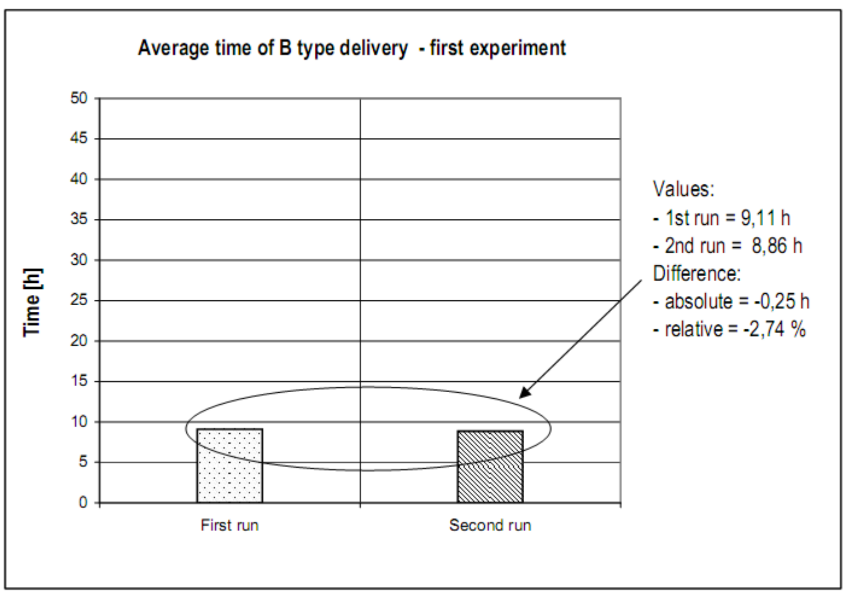

Figure 21. Average B type delivery's time in the first experiment

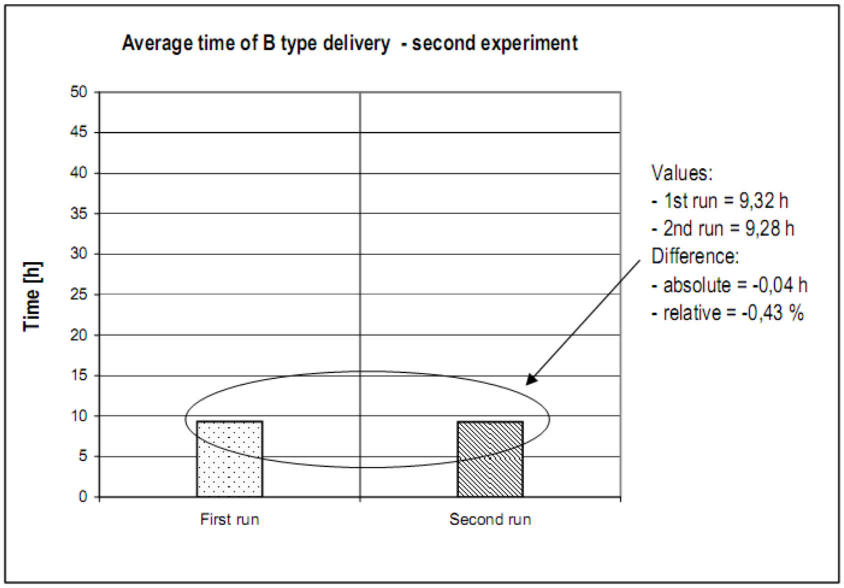

Figure 22. Average B type delivery's time in the second experiment

Figures 23. and 24. show C category deliveries, which consist of the following stages:

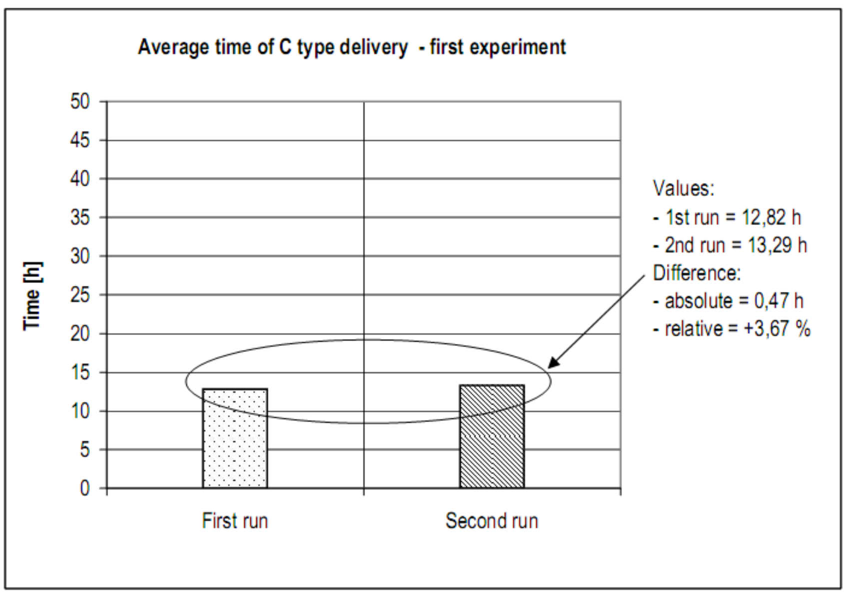

Figure 23. Average C type delivery's time in the first experiment.

1. parts production (supplier's production process),
2. delivery of parts (between supplier and manufacturer),
3. transformation of parts into final products (manufacturer's production process),
4. delivery of products (between manufacturer and retailer).

Average delivery time in this case equals approximately 13 hours in the first experiment and 14.5 hour in the second experiment. Simulated bullwhip effect has not had an important influence on B and C categories.

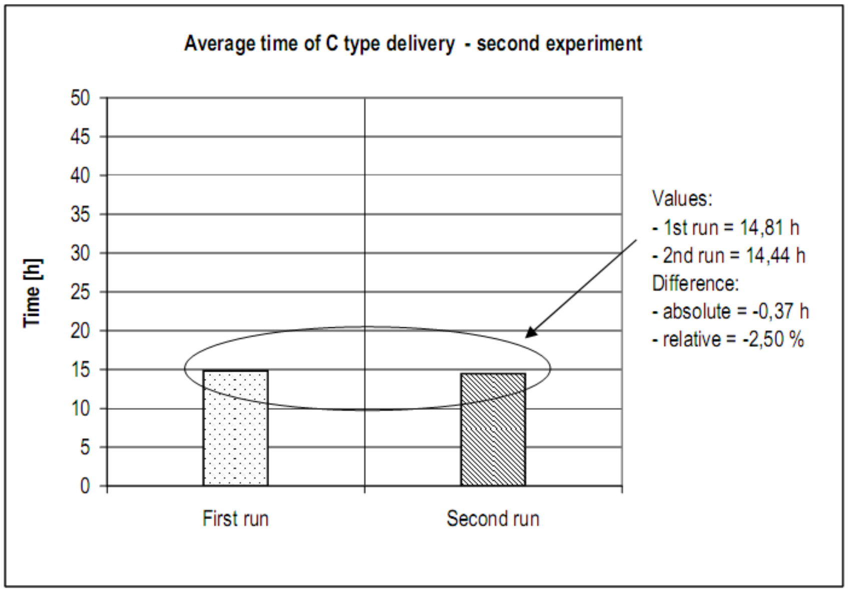

Figure 24. Average C type delivery's time in the second experiment

Figures 25. and 26. show very interesting and spectacular results. Both graphs illustrate the longest delivery time in whole simulation's process. In the first experiment difference between longest delivery time (if the bullwhip effect was simulated or not) is equal 10.2 hours, so it means 39.53%. In the second experiment the difference was even bigger: 12.3 hours – 47.49%. To prove that randomly generated demand pattern did not have important influence on the results, it is worth to compare longest delivery time in first run in both simulation's experiments, which is about 26 hours.

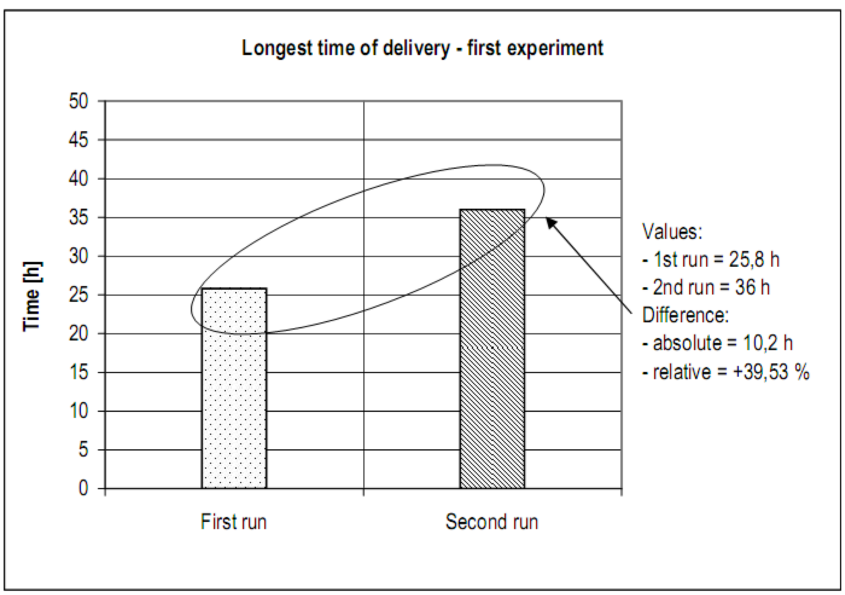

Figure 25. The longest delivery time in the first experiment

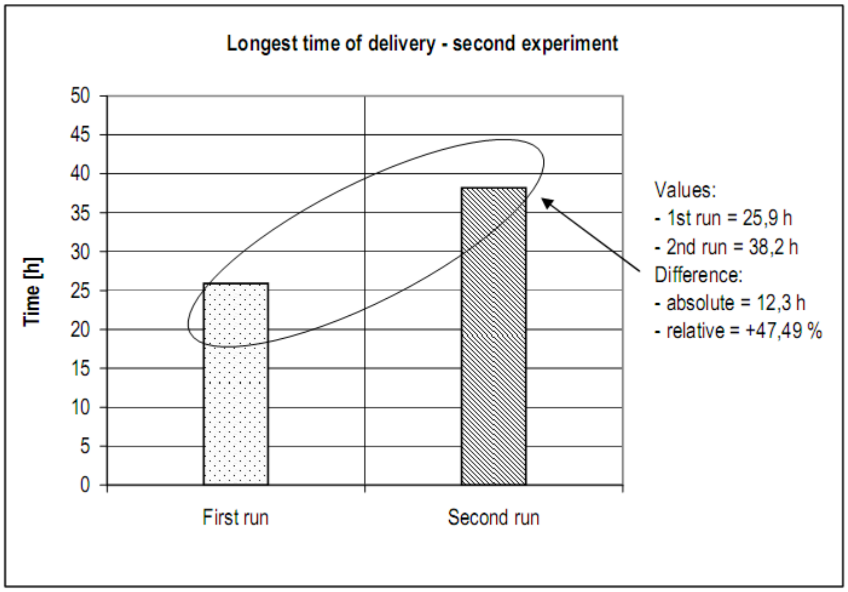

Figure 26. The longest delivery time in the second experiment

To conclude there is no doubt that the bullwhip effect causes longer delivery time, it is clearly shown on the figures 25 and 26. In the implemented simulation, in both experiments and both runs, bullwhip effect was simulated only once during simulation's process (which was quite long). We can suppose that if bullwhip effect occurs more frequently, negative effect of it will be bigger and much more serious. To prove this thesis, another research should be conducted with 30 – 50 simulation cycles, and number of bullwhip effect simulations should be changed from 1 to different values.

The experiment has been focused on the time dimension of bullwhip effect influence on supply chain. Of course is it possible to examine different aspects of the supply chain operation such as: productivity of very long and very short supply chains, influence of bullwhip effect on production stability or money resources.

Conclusions

The nature of business organizations and characteristics of their environments have changed. More flexible and sensitive constructs are needed to deal with modeling of business architectures which operate rather on social principles than on mechanistic rules. What is more, because of complexity of such structures as Extended Enterprises, simulation seems to be only reasonable analysis method for understanding existing business models as well as designing new ones. Simulation offers an effective analytical tool for organizations that need to understand their behavior and measure the performance of cycle time in the environment of Extended Enterprise. As is shown in a paper agent orientation is powerful paradigm for modeling contemporary enterprises and conducting simulation experiments. Proposed reference model provides modeler with step-by-step process as well as detailed structure and shows how to develop simulation model and experiment with it. The reference model elements have been implemented in NetLogo environment and used during supply chain analysis described in the case study. The framework is not claimed to be exhaustive or complete. It is intended to be a core on which further research will be done.

References

[Bansal, 2002] Bansal S.: Promise and problems of simulation technology in SCM domains. Proceedings of the 2002 Winter Simulation Conference, (2002), pg. 1831-1837.

[Davenport, 1995] Davenport T.: Process Innovation. Reengineering Work Through Information Technology. Harvard Business Press, (1995).

[Davidsson, 2002] Davidsson P.: Agent based social simulation: a computer science view, J. Artif. Soc. Social Simulation, 5, (2002), avaiable at http://jasss.soc.surrey.ac.uk/5/1/7.html.

[Davidsson, 2000] Davidsson P.: Multi agent based simulation: beyond social simulation. In: Multi Agent Based Simulation, S. J. Moss and P. Davidsson, eds., Springer-Verlag LNCS series, vol. 1979, (2000).

[Eriksson et al., 2000] Eriksson H. E., Penker M.: Business modeling with UML: Business patterns at work. John Wiley & Sons, (2000).

[Fettke et al., 2006] Fettke P., Loos P.: Reference Modeling for Business Systems Analysis. IGI Global, (2006).

[Forrester, 1961] Forrester J. W.: Industrial Dynamics. Cambridge, (1961).

[Forrester, 1968] Forrester J. W.: Principles of Systems, Cambridge, (1968).

[Genesereth et al., 1987] Genesereth M. R., Nilsson N. J.: Logical Foundations of Artificial Intelligence. Morgan Kaufmann, (1987).

[Gilbert, 2007] Gilbert N.: Agent-Based Models, Sage Publications, (2007).

[Hammer, 1990] Hammer M.: Reengineering Work: Don't Automate, Obliterate. Harvard Business Review, (1990) pg.104-113.

[Holland, 1975] Holland J. H.: Adaptation in natural and artificial systems, Cambridge, (1975).

[Inverno et al., 1997] Inverno M. de, Fisher M., Lomuscio A., Luck M., Rijke M. de, Ryan M., Woolridge M. : Formalisms for Multi-Agent Systems. In Knowledge Engineering Review, vol 12, No. 3, (1997).

[IT GI, 2005] IT Governance Institute: Governance of the Extended Enterprise: Bridging Business and IT Strategies. John Wiley & Sons, (2005).

[Jakieła, 2006] Jakieła, J.: AROMA – Agentowo zoRientowana metOdologia Modelowania orgAnizacji. WAEiI, Politechnika Śląska, Gliwice (2006)

[Jakieła, Pomianek, 2009] Jakieła J., Pomianek B.: Agent Orientation as a Toolbox for Organizational Modeling and Performance Improvement. Intenational Book Series "Information Science and Computing", Book 13, Intelligent Information and Engineering Systems, INFOS 2009, pp. 113-124, (2009).

[Jakieła, Litwin, Olech, 2010] Jakieła J., Litwin P., Olech M.: MAS Approach to Business Models Simulations: Supply Chain Management Case Study. In: KES AMSTA-2010, Jędrzejowicz P. et al. (Eds.), Part II, LNAI 6071, pp. 32-41, Springer-Verlag, Berlin Heidelberg, (2010).

[Luck, 2003] Luck, M., McBurney, P., Preist, C.: Agent technology: enabling next generation computing. A roadmap for agent based computing, (2003), www.agentlink.org.

[McCormack et al., 2003] McCormack, K., Johnson, W.C.: Supply chain networks and business process orientation. Advanced Strategies and Best Practices, CRC Press LLC, Florida (2003).

[North et al., 2007] North, M.J., Macal, C.M.: Managing Business Complexity. Discovering Strategic Solutions with Agent-Based Modeling and Simulation. Oxford University Press (2007).

[Nuno et al., 2009] Nuno D., Sichman J. S. (Eds.): Multi-Agent-Based Simulation IX. Springer-Verlag, Berlin Heidelberg, (2009).

[Padgham et al., 2004] Padgham L., Winikoff M.: Developing Intelligent Agent Systems: A Practical Guide. Wiley & Sons, (2004).

[Paolucci et al., 2005] Paolucci M., Sacile R.: Agent-Based Manufacturing and Control Systems. New Agile Manufacturing Solutions for Achieving Peak Performance. CRC Press LLC, (2005).

[Saleh, 2000] Saleh M.: The Hard Core of the System Dynamics Research. In Davidsen Pal I., David N. Ford and Ali N. Mashayeki (EDs.): Sustainability in the Third Millenium, Bergen, (2000).

[Shieritz et al., 2003] Shieritz N., Miling P. M.: Modeling the Forest or Modeling the Trees. A Comparison of System Dynamics and Agent-Based Simulation. In Proceedings of 21 System Dynamics Conference, New York, (2003).

[Smith, 1776] Smith A.: The Wealth of Nations, 1776, in Shafritz J. M., Ott J. S.: Classics of Organization Theory. Cole Publishing Company, (1992).

[Vieira et al. 2005] Vieira, G.E., Cesar, O. Jr.: A conceptual model for the creation of supply chains models. Proceedings of the 37th conference on Winter simulation, pp. 2619 – 2627, Orlando, Florida, (2005).

[Weyns et al., 2009] Weyns, D., Uhrmacher, A.M. (eds.): Multi-Agent Systems Simulation and Applications. Computational Analysis, Synthesis, and Design of Dynamic Models Series. CRC Press, Florida (2009)

[Yilmaz et al., 2009] Yilmaz L., Ören T.: Agent-Directed Simulation and Systems Engineering, John Wiley & Sons, (2009).

Authors' Information

Jacek Jakieła, Ph.D., Eng. *– Department of Computer Science FMEA RUT; W. Pola 2, 35-959 Rzeszow, Poland; e-mail: jjakiela@prz.edu.pl*

Major Fields of Scientific Research: Software Development Methodologies, Agent and Object-Oriented Business Modeling, Internet Enterprises Models, Computational Organization Theory and Multi-Agent Based Simulation of Business Architectures.

Paweł Litwin, Ph.D., Eng. *– Department of Computer Science FMEA RUT; W. Pola 2, 35-959 Rzeszow, Poland; e-mail: plitwin@prz.edu.pl*

Major Fields of Scientific Research: Applications of Neural Networks in Mechanics, Computer Simulations, Finite Element Method.

Marcin Olech, M.Phil., Eng. *– Department of Computer Science FMEA RUT; W. Pola 2, 35-959 Rzeszow, Poland; e-mail: molech@prz.edu.pl*

Major Fields of Scientific Research: Multi-Agent Based Simulations, Application of Artificial Intelligence in Industry.