CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Agent-Based Simulations for Advanced Supply Chain Planning: The FAMASS Methodological Framework for Requirements Analysis

Luis Antonio de Santa-Eulalia
Sophie D'Amours
Jean-Marc Frayret

April 2011

CIRRELT-2011-22

Bureaux de Montréal :

Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :

Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

Agent-Based Simulations for Advanced Supply Chain Planning : The FAMASS Methodological Framework for Requirements Analysis

Luis Antonio de Santa-Eulalia[1,*], Sophie D'Amours[2,3], Jean-Marc Frayret[2,4]

[1] TÉLUQ, Université du Québec à Montréal, 455, rue du Parvis, Québec, Canada G1K 9H6

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[3] Department of Mechanical Engineering, Université Laval, Pavillon Adrien-Pouliot, 1065, avenue de la médecine, Québec, Canada G1V 0A6

[4] Mathematics and Industrial Engineering Department, École Polytechnique de Montréal, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7

Abstract. Agent-based systems have been employed in the Supply Chain Management field since the 1990s. In spite of its appealing and extensive use in both research and practice, the agent technology and its integration with distributed supply chain planning tools still represent an emergent field with many open research questions. Particularly, the literature fails to provide an integrated framework to identify, model and conduct simulation experiments covering the whole simulation cycle. Indeed, the initial modelling effort performed at the analysis phase is especially neglected by the literature concerned. This early phase is critical because it considerably influences the whole development process as well as the resulting simulation experiments. Thus, this paper presents a novel methodological framework called FAMASS (*F*ORAC *A*rchitecture for *M*odelling *A*gent-based *S*imulation for *S*upply chain planning), which provides: i) a uniform representation of distributed advanced supply chain planning systems using agent technology; and ii) a methodological approach supporting analysts in defining functional requirements of possible simulation experiments. This paper introduces FAMASS and presents a proof-of-concept based on a real-scale industrial application.

Keywords. Agent-based simulation, analysis modelling, supply chain management, advanced supply chain planning systems, methodological framework.

Acknowledgements. The authors wish to thank the Natural Sciences and Engineering Council of Canada (NSERC) and the FORAC Research Consortium (www.forac.ulaval.ca) for their financial support, as well as the two anonymous referees for their valuable recommendations to improve the manuscript.

---

* Corresponding author: leulalia@teluq.uqam.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2011

1 Introduction

Supply chain planning is an important and complex business process. It aims to obtain a balance between supply and demand, from suppliers to customers, in order to deliver superior goods and services through the optimization of supply chain assets. This is quite a difficult task since it involves a large quantity of complex decisions to be synchronized.

To cope with the complexity of the supply chain planning process, decision support tools have been developed since the last decade (Shapiro 2000). Perhaps one of the most prominent approaches in this area concerns supply chain planning systems employing agent technology. This technology is able to capture the distributed nature of the supply entities (e.g., customers, manufacturers, logistic operators, etc.) to mimic their business behaviours and support the collaborative planning process of the supply chain entities. Because of these abilities, among several others described in the literature, agent-based supply chain systems have great potential for simulating complex and realistic scenarios (Lee and Kim 2008).

Although several attempts have been made to take advantage of agent technology to specify, design and implement agent-based supply chain simulation, the related literature does not address the analysis phase in detail (Santa Eulalia et al. 2008; Govindu and Chinnam 2006; Galland et al. 2003). The analysis is the first development phase, where functionalities of the system and non-functional constraints have to be described. This phase is of crucial relevance (Fontanilli 2008) because it helps determine the requirements of the simulation experiments. At this phase the simulation problem is clearly defined according to the available theory, so that it can be translated into simulation requirements, for example by identifying experimental factors and responses. To our knowledge, there are no specialized methodological frameworks for analyzing simulations in the context of supply chain planning. Due to the complexity of supply chains systems, analysts are confronted with a huge amount of combinations that form different experimental scenarios to be tested and, consequently, a vast array of possible simulation requirements.

Furthermore, Robinson (2008) explains that this initial phase (also known as the conceptual modelling) is usually seen as the most essential part of any simulation study, but it is also the least understood. Among the research opportunities pointed out by the author, he highlights the need to understand how software engineering techniques might aid in conceptual modelling; developing appropriate model representation methods; identifying, adapting and developing conceptual modelling frameworks; and understanding how to organize and structure the knowledge gained during modelling.

In order to contribute to reducing this research gap, this work proposes a methodological framework called FAMASS (*F*ORAC *A*rchitecture for *M*odelling *A*gent-based *S*imulation for *S*upply chain planning) for the analysis phase of the development of agent-based advanced supply chain planning simulations. The FAMASS approach provides a uniform representation of distributed supply chain planning systems using agent technology to support simulation analysts in defining what the functional requirements of possible simulation scenarios are. The proposed methodological framework was tested though a proof-of-concept case for the forest products industry.

The paper is organized into the following sections: Section 2 provides a literature review and explains in more details the research gap. Next, Section 3 presents the FAMASS methodological framework. Section 4 provides a proof-of-concept case application. Finally, Section 5 states some final remarks and suggests future work.

2 Related Works

Due do the distributed nature of supply chains, agent-based systems are of great utility in helping solve supply chain planning problems (Lee and Kim 2008; Frayret et al. 2008, Monteiro et al. 2008). An "agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives" (Wooldridge, 1998). When modelling a supply chain, different agents together form what is called a multi-agent system (a society of agents). A multi-agent system is defined as a set of agents that work together and interact with one another in order to accomplish certain tasks. All of them use their competence and knowledge to strengthen the capacity of the entire planning system in solving problems. When agents of a supply chain planning system employ optimization technology, they can be defined as distributed APS (Advanced Planning and Scheduling) systems (Santa-Eulalia et al., 2008).

Since the 1990s, the literature has provided a set of interesting approaches for modelling agent-based systems for supply chain planning. The literature can be divided into two types of contributions: I) agent-based supply chain management systems, where agents are dedicated to supply chain management, but are not specialized in the advanced supply chain planning domain, i.e., they do not mention the use of optimization or finite capacity planning approaches for supply chain planning; II) agent-based advanced supply chain planning systems, which explicitly mention the use of optimization procedures or finite capacity planning models and techniques.

Examples of relevant projects in the first category are Karam et al. (2010), Labarthe et al. (2007), Chatfield et al. (2007), Govindu and Chinnam (2006), and Van der Zee and Van der Vorst (2005). For instance, Karam et al. (2010) present an organization-oriented methodological framework for modelling and simulation of the impact of a specific supply chain organizational structure and its related management policies on supply chain performance. Labarthe et al. (2007) propose an approach for modelling customer-centric supply chains in the context of mass customization. Govindu and Chinnam (2006) propose a method for the analysis and design of multi-agent supply chain systems by integrating the Gaia methodology and SCOR (Supply-Chain Operations Reference) model. Van der Zee and Van der Vorst (2005) put forward an agent approach derived from an object-oriented approach to explicitly model control structures of supply chains.

In terms of the second category, several interesting researches exist, including DIMA (Ivanov 2009), Egri and Vancza (2005), SNS (Baumgaertel and John 2003), ANTS (Sauter et al. 1999), and Swaminathan et al. (1998). For instance, the DIMA (Decentralized Integrated Modelling Approach) (Ivanov 2009) introduces a new conceptual architecture for multi-disciplinary modelling of structural planning and operations of adaptive supply chain with dynamics considerations. Egri and Vancza (2005) is a Gaia-based approach for modelling advanced distributed supply chain planning for mass customization. Baumgaertel and John (2003) propose an agent-based simulation architecture for supply networks, incorporating Advanced Planning and Scheduling components and using finite domain constraint technology. The Sauter et al. (1999) architecture, called ANTS (Agent Network for Task Scheduling), consists of a supply chain planning system composed of agents inspired by human intuitions and insect colonies. Swaminathan et al. (1998) provide a supply chain modelling framework containing a library of modular and reusable software components, which are different kinds of supply chain agents, their constituent control elements and their interaction protocols. A detailed and recent comparative discussion about agent-based systems for supply chain management can be found in Monteiro et al. (2008).

All these works have greatly contributed to the domain of agent-based frameworks for advanced supply chain planning. Nevertheless, in spite of these advances, some research gaps still exist (Santa-Eulalia 2009). We would like to draw attention to two of them:

- Most of the frameworks do not explicitly explore the concept of advanced planning and scheduling. Opportunities exist to discuss how agent-based technology can be used to capture the planning mechanisms at the strategic, tactic and operational levels, from the source of raw materials to the final consumption.
- Most of the frameworks propose a modelling shell to specify, design and implement agent-based advanced planning systems. However, none of them, to our knowledge, explores how to identify a problem in this domain and how it can be translated into simulation requirements, for example by identifying experimental factors and responses.

Both of these gaps are related to the 'analysis phase' of the design of agent-based simulation environments, which involves the comprehensive examination of the problem to be solved. This analysis represents the first development phase, where the functions of the system and non-functional constraints have to be described. This phase is of crucial relevance (Fontanili et al. 2008), because it allows simulation analysts to understand what the system can model according to the available theory, as well as how a given supply chain planning problem can be translated into a simulation platform.

Thus, to help fill this research gap, the next subsection contextualizes the main contribution of this work.

2.1 Contribution of the FAMASS Approach

To position FAMASS in relation to existing works, Figure 1 organizes the published literature according to two mains axes: the ‘modelling view’ and the ‘methodological phases’, of the development process. This will also later facilitate the explanation of the proposed FAMASS approach. The ‘modelling view’ and the ‘methodological phases’ are respectively represented as the vertical and horizontal axes.

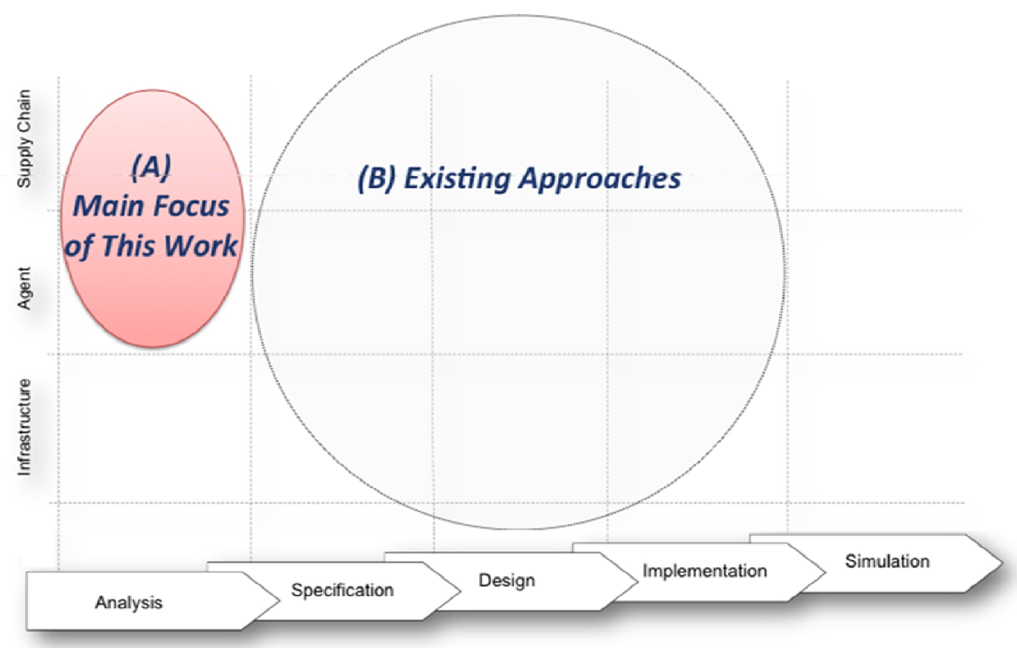

Figure 1: The FAMASS contribution and the published literature.

The modelling view (vertical axis) comprises the Supply Chain (i.e. the business viewpoint), the Agent (i.e. the supply chain domain problem translated into an agent-based view) and the Infrastructure (i.e. how an agent system can be supported by computing resources, such as integrating infrastructure and hardware).

As for the methodological phases, the grid in Figure 1 adheres to the methodology for simulation of distributed systems developed by Galland et al. (2003). The Analysis phase performs an abstract description of the modelled supply chain planning system containing the simulation requirements. During this phase, the functionalities of simulation are identified and described in general terms. Specification translates the information derived from the analysis into a formal model. As the Analysis phase does not necessarily allow obtaining a formal model, the Specification examines the analysis requirements and builds a model based on a formal approach. The Design creates a data-processing model that describes in more detail the specification model. In the case of an agent-based system, design models are close to how agents operate. Implementation translates the model resulting from the Design phase into a specific software platform, such as ARENA®, REPAST®, or AnyLogic®. Simulation stands for the use of the simulation model by customers according to a set of experimental plans

The grid in Figure 1 is used to organize the modelling approaches. It does not mean that one has to cover the whole grid when modelling supply chain systems, but rather represents a way to arrange all possible modelling approaches for a study domain and understand how they are related.

Most of the existing approaches for agent-based advanced supply chain planning are for the Specification and Design (represented by the ‘B’ area) phases. Figure 1 indicates that the present work focuses on the analysis phase for the ‘supply chain’ and ‘agent’ views, as represented by the ‘A’ area. In the next section, four interactive modelling approaches are proposed to cover this area.

3 The FAMASS Methodological Framework

This section presents the main contribution of this paper: the FAMASS (*F*ORAC *A*rchitecture for *M*odelling *A*gent-based *S*imulation for *S*upply chain planning). The term architecture is used here as a synonym of framework, although the literature slightly differentiates between the two terms (Vernadat 1996).

FAMASS comprises four interactive modelling approaches, as schematized in Figure 2. At the supply chain level, they are: GPA (General Problem Analysis) and DPA (Distributed Problem Analysis). At the agent level, they are: SAOA (Social Agent Organization Analysis) and IAOA (Individual Agent Organization Analysis).

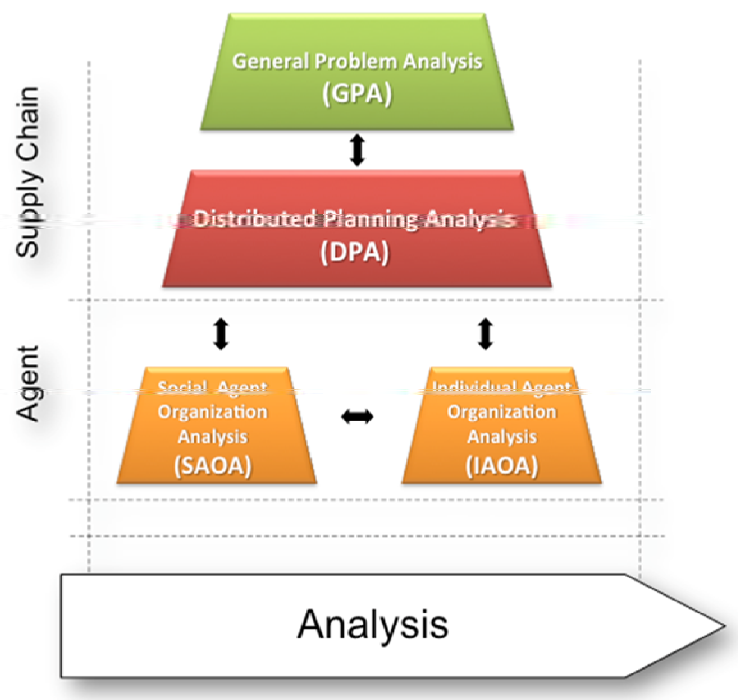

Figure 2: Four main modelling approaches proposed for analysis of supply chain and agent levels for simulation purposes.

The proposed four approaches are part of the FAMASS methodology and should be driven step-by-step. They will be explained in subsection 3.2. Before presenting them, it is necessary to explain that we propose two basic activities for each modelling approach at the analysis phase: requirements determination and requirements structuring (Hoffer et al. 2005).

3.1 The Analysis Phase

The first analysis activity is requirements determination, as explained next.

3.1.1 Requirements Determination

During requirements determination, simulation analysts identify what the requirements are according to the views of the simulation stakeholders. In order to do so, four metamodels are proposed, one for each modelling approach in Figure 2. The proposed metamodels are domain guidelines on what type of information has to be discussed, serving as a reference to determine what the possible simulation requirements shall be. Basically, the proposed metamodels translate the foundations of advanced planning and scheduling into a set of agent-based concepts that can be useful in the supply chain planning context. The metamodels serve as a superstructure or a concept map that aims at organizing both the terminology to be used and the structure of concepts employed when thinking in terms of a model's content.

After determining the basic simulation requirements, they have to be organized.

3.1.2 Requirements Structuring

Requirements structuring stands for a coherent representation of the information gathered through diagrams. Typical requirement structuring methods include data flow diagrams and use case diagrams (Hoffer et al. 2005).

FAMASS suggests adapted versions of use case diagrams and requirements diagrams from SysML (OMG 2010). As an extension of a subset of the UML (Unified Modelling Language), the SysML (Systems Modelling Language) is the most recent open-source initiative from the Object Management Group (OMG). It is basically a general-purpose modelling language for systems engineering, supporting the analysis, specification, design, verification and validation of a broad range of systems and systems-of-systems.

FAMASS also employs agent-based use cases from AUML (Agent UML) (Heinze at al. 2000). The reasons for using a set of UML-compliant approaches are related to the fact that UML is considered interesting, by the simulation community, in order to represent a conceptual model (Robinson 2004); UML inspired the creation of a unified language for enterprise modelling (Vernadat 2002); it is being applied in the supply chain management area (e.g. Derrouiche et al. 2008; Chu et al. 2008); and the UML community is pushing to standardize AUML within the FIPA – Foundation for Intelligent Physical Agents (Huget 2004).

The following subsection presents the four modelling approaches.

3.2 FAMASS's Four Modelling Approaches

The proposed four modelling approaches are schematized in Figure 2 and now explained.

3.2.1 General Problem Analysis (GPA)

GPA is the first modelling effort where simulation analysts have to think about the simulation problems. The GPA is created based on Santa-Eulalia et al. (2008). The metamodel for the GPA proposes that the simulation analysis has to take into consideration two main aspects: general aspects and experimental aspects. General aspects represent macro definitions of the simulation problem, including the object and environment to be simulated, the simulation questions and objectives. Experimental aspects are related to the design of experiment, where one defines the factors, uncertainties and key performance indicators. These issues are summarized in what follows.

General Aspects

Simulation Object: simulation is about empirically testing the performance of an object in a certain environment. In this case, the object is something logical or physical that is perceived by decision-makers and constitutes the subject matter of an investigation. For example, the simulation object can be something physical, such as new technologies to be employed in the distribution centre, or something logical, such as different decoupling points in a supply chain. The decoupling points are inventory locations permitting the connection between push-pull systems.

Simulation Environment: represents the circumstances or conditions by which the object is surrounded and which ultimately influence the performance of the simulation object. For example, the experiment would be to configure different decoupling points (simulation object) in a planning and control system dedicated to the Canadian lumber supply chain (simulation environment). In this case, the simulation environment is composed of a decision system (i.e. a planning and control system) and a dataset (from the Lumber industry) providing the context for the supply chain, including demand, supply and manufacturing information. Thus, the object defines what the analysts desire to study and the environment is all else that is included in the simulation, i.e. the entire context.

Simulation Questions: represent the interrogations simulation analysts have concerning the performance of the object under a certain environment. For example, one would like to investigate whether the performance of a decoupling point position close to the client is superior to one closer to the supply source. Normally, these questions are related to simulation hypotheses.

Simulation Objective: defines the purpose or reason for conducting the simulation. Influenced by Harrell et al. (2004), six objectives categories are proposed to facilitate identifying the objective: i) Performance analysis: defines what the performance of the system (or part of the system) is in terms of resource utilization, flow time, output rate, etc.; ii) Capacity/constraint analysis: aims to determine what the system performance is when pushed to the maximum and what limits the system (i.e., bottleneck); iii) Optimization: determines which combination of particular decision variables best achieves desired performance metrics; iv) Sensitivity analysis: studies which decision variables are most influential on performance metrics and how influential they are; v) Visualization/demonstration: determines how the dynamics of the system can be more effectively visualized and/or demonstrated; vi) Trade-off analysis: simulation can be created with the objective of studying trade-offs, i.e. for investigating the balance of factors which are not all achievable at the same time (e.g. part of a supply chain can go well, while part of it crashes). By understanding these objectives categories, the simulation objectives become clearer. A simulation study can have one or many objectives, which may change or even expand as the simulation project advances.

By knowing the scope of the simulation and its corresponding objectives, it becomes easier to define the experimental aspects of the simulation.

Experimental Aspects

Factors: stand for controllable decision variables or policies with which managers can work. Factors have levels, i.e. a rank or degree. For example, from the General Aspects previously explained, one could state that ‘two’ different decoupling points (levels) have to be tested.

Uncertainties: are uncontrollable factors or non-policy variables, which cannot be controlled by managers, but can be controlled by the analyst during the experiments. Uncontrollable factors are uncertain and can be seen as noises or variations in the simulation object or in its environment. For example, managers cannot control demand variability, but it is of great importance for supply chain performance. Although managers very often cannot control the demand, the simulation system can be tested according to different demand variability.

Key Performance Indicators (KPI): stand for experimental responses or observations about the simulation performance.

These three issues (factors, uncertainties and key performance indicators) are quite important when defining a simulation experiment. Of course, they can be related to the supply chain level (i.e. the business level) like the decoupling point position, but they can also be related to the agent level (see Figure 2). For example, different social structures of the agent society (e.g. a hierarchical or heterarchical structure) could be factors; stochastic algorithms for coalition formation could introduce uncertainties in the system; and the performance of this system might be measured through the quantity of agents created during a simulation round. These issues will be clearer when the modelling approaches for the agent level are introduced in subsections 3.2.3 and 3.2.4.

All the above-mentioned GPA aspects guide the simulation team to generate, through interviews, a set of atomic requirements that will guide the remaining modelling effort. These requirements are normally produced in a text format. As the quantity of requirements can be rather great for some situations, it would be interesting to organize all of them through a requirements structuring approach. For instance, the FAMASS approach employs requirements diagrams from SysML (OMG, 2010). An example is presented later to illustrate the GPA effort.

The remaining steps of the FAMASS methodology clarify the problem definition and the general requirements of the GPA, as it is discussed next.

3.2.2 Distributed Planning Analysis (DPA)

The DPA identifies what the desired supply chain planning entities are, as well as their roles. These entities are identified according to their mission in the supply chain and their planning functions at different decision levels.

To identify them, we employ the concepts of supply chain integration proposed by Shapiro (2000). The author states that supply chain management refers to integrated planning relying on three basic dimensions: i) *Spatial dimension*: refers to the fact that supply chains are composed of geographically dispersed units of analysis; ii) *Functional dimension*: stands for different planning functions in a supply chain, which can be related to procurement, manufacturing, distribution and sales; iii) *Intertemporal dimension*: refers to different decision levels, i.e., strategic, tactical and operational decision levels.

From these dimensions emerges the notion of a Supply Chain Block (SCB). A SCB can be defined as a supply chain planning entity, which is a functional unit capable of performing part of the supply chain planning processes or its totality, or part of the execution of the supply chain decisions or its totality. These entities have a certain degree of autonomy and are able to interact with each other. Possible Supply Chain Blocks for covering the integrated supply chain planning dimensions of Shapiro (2000) are proposed in the framework of Figure 3, which is called the ‘supply chain planning cube’.

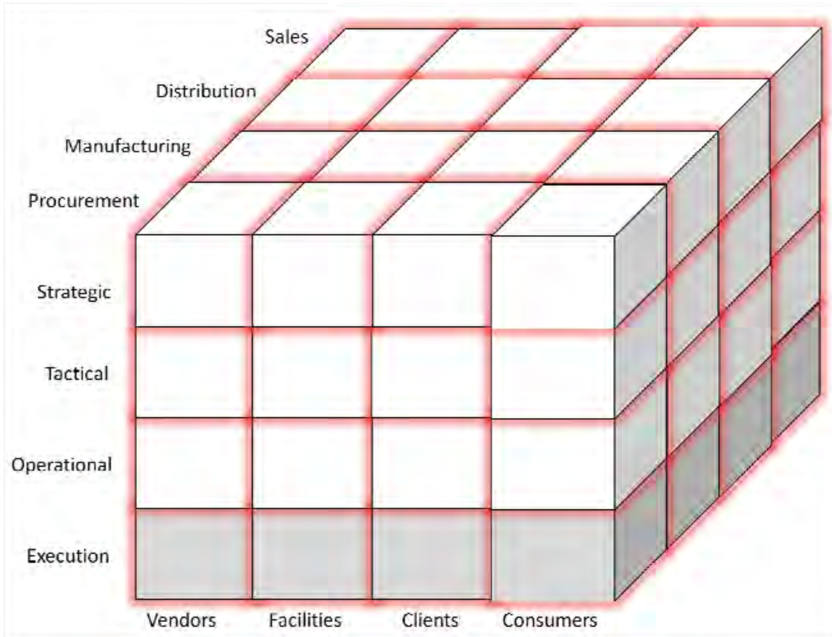

Figure 3: Supply chain planning cube.

The proposed supply chain planning cube identifies the possible planning and control functions of a typical supply chain. For example, the Supply Chain Block responsible for 'manufacturing – operational – facilities' is in charge of the operational planning activities (i.e., short-term planning and scheduling) for manufacturing facilities. If one desires to include the operational planning for facilities responsible for distribution (e.g. a distribution centre), the 'distribution – operational – facilities' block can be plugged into the previous block.

A vertical slice of the supply chain planning cube for one spatial unit of analysis (e.g. facilities) is similar to the planning matrix proposed by Meyr and Stadtler (2004), except for the execution level. Figure 4 schematizes a typical APS system as defined by Meyr and Stadtler (2004) using the supply chain planning cube.

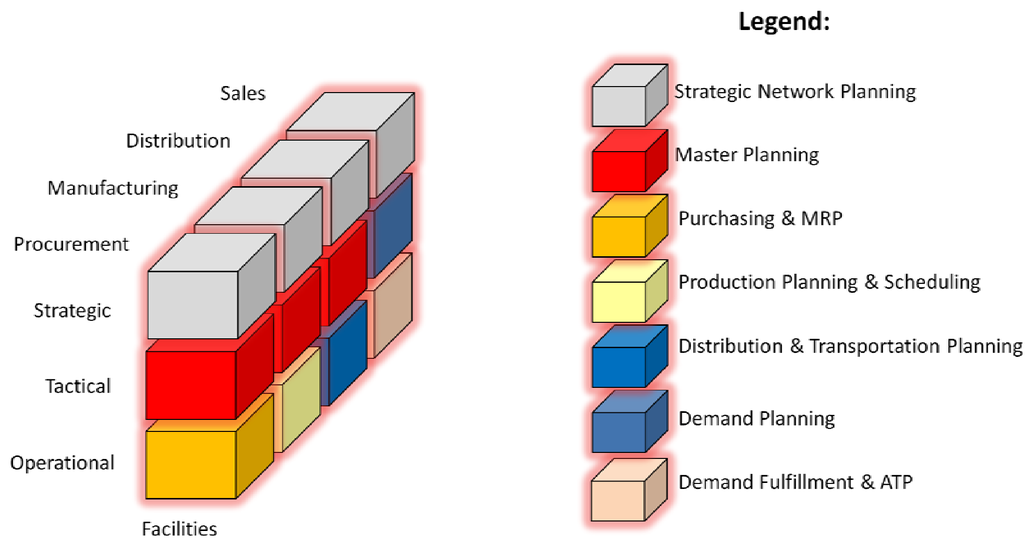

Figure 4: The Supply Chain Matrix of Meyr and Stadtler (2004) APS organized according to the supply chain planning cube.

In this sense, the supply chain planning cube is an evolution of the planning matrix, due to the fact that it represents the possibility of collaboration among different traditional APS systems. It also goes further by including execution entities.

It is interesting to note that a well-known specialized framework from the literature, the SCOR model (SCOR 2006), can also be covered by this supply chain planning cube, as represented in Figure 5.

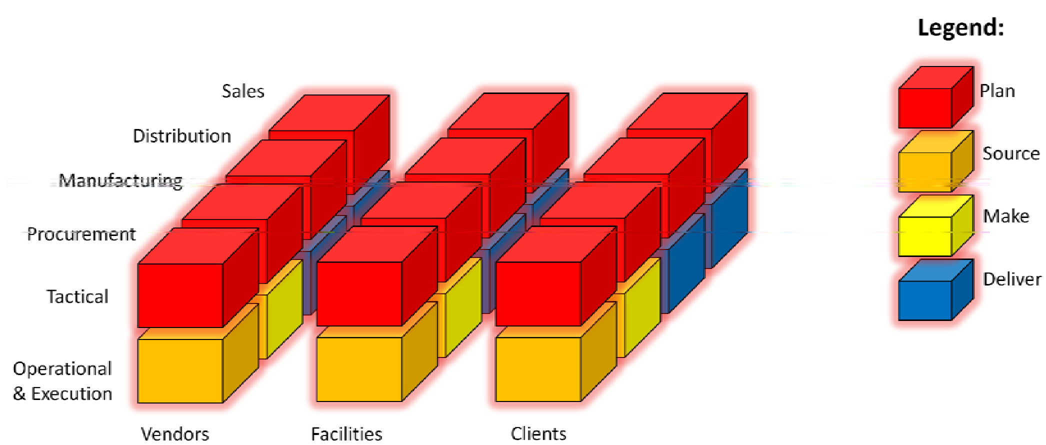

Figure 5: The SCOR model organized according to the supply chain planning cube.

The SCOR model proposes five basic supply chain entities (plan, source, make, deliver and return), and four of them are represented in Figure 5[1]. Figure 5 presents three SCOR models side-by-side to represent a three-echelon supply chain comprising vendors, facilities and clients.

When using the supply chain planning cube, it is important to note that the ‘facilities’ slice represents the central company and its internal supply chain. In addition, a Supply Chain Block can be combined with other Supply Chain Blocks or it can be decomposed. An example of a combination is represented by the ‘plan’ function in Figure 5 for the SCOR model, which is composed of several red blocks. In terms of decomposition, any entity can be disaggregated into more elementary entities if desired. For example, Montreuil et al. (1995) explain that a production entity (in this case, an execution Supply Chain Block for manufacturing) can be decomposed into work centres, work zones, or work stations. We decided to keep the same granularity level of Meyr and Stadtler (2004) and Shapiro (2001), but analysts are offered the possibility of manipulating the cube according to their needs.

Based on the supply chain cube, one has to perform requirements determination for the simulation aspects. Just like the GPA, a set of atomic requirements are gathered through interviews with simulation stakeholders for the DPA. Next, the resulting atomic requirements have to be organized through requirements structuring. FAMASS proposes use cases combined with requirements diagrams from SysML. Section 4 illustrates how these requirements structuring approaches are employed.

[1] The supply chain cube can be extended to take into account: return, refurbishing, recycling and reversed logistics if needed, by extending the “Z” axis of the cube.

3.2.3 Social Agent Organization Analysis (SAOA)

So far, the concept of Supply Chain Block has been used to represent entities responsible for part of the supply chain planning. Together, they compose a population of Supply Chain Blocks interacting with each other, having a collective co-existence within the planning system. When these entities incorporate attitudes, orientations and behaviours comprising the interests, needs or intentions of another Supply Chain Block, they can be seen as social entities. A way to represent social entities is to model them as agents, thus creating multi-agent societies.

In this regard, the objective of the SAOA is to translate the DPA model into a multi-agent society. The main functional entities from the DPA are the Supply Chain Blocks, which are not necessarily directly related to agents. The notion of agent goes beyond the notion of Supply Chain Blocks, consequently, an explanation about agentification (i.e. how to transform Supply Chain Blocks into agents) is necessary at this point.

The simplest situation is where a Supply Chain Block can be directly transformed into an agent. The left side of Figure 6 indicates that a Supply Chain Block can be encapsulated as a simple agent, where the frontier between a Supply Chain Block and an agent is delimited by the definition of agents. A Supply Chain Block can become an agent when properties of agents are introduced, for example, when pro-activeness is introduced into a Supply Chain Block. As an agent moves from left to right, as illustrated in Figure 6, new layers that represent additional human capacities are added, like sophisticated social abilities, learning capacity, etc. The most complex agents in a supply chain are probably human agents.

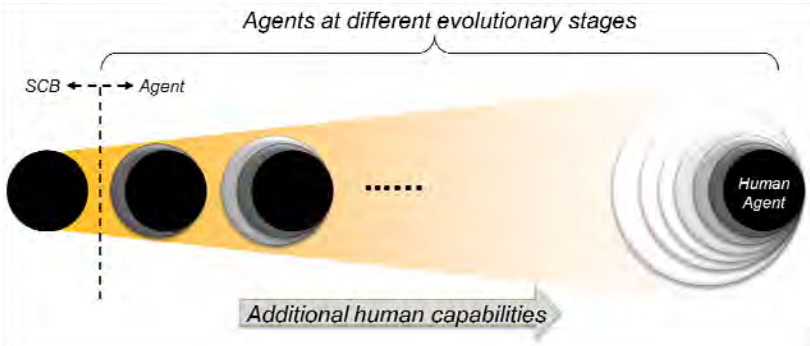

Figure 6: The agentification process.

This idea leads us to naturally think that a Supply Chain Block can be directly translated into an agent by adding agent abilities to it. This is based on the agentification definition of Shen et al. (2001), who explain that the agentification process can be functional-based (i.e. the white Supply Chain Block in Figure 3) or physical-based (i.e. the grey Supply Chain Block in Figure 3).

However, a Supply Chain Block can sometimes be transformed into more than one agent, for example when specialization is required. In this case, a planning agent can be specialized according to generic responsibility orientations (Montreuil and Lefrançois 1996) such as products, processors, processes or projects, in order to obtain faster or more precise responses for specific given situations. In other cases, different intermediary agents can be created to perform activities related to e.g. the coordination of the agents' society. In addition, the agentification process can also include the representation of information sources, interfaces and other services.

The importance of this discussion relies on the notion that agentification is the basis for two mutually dependent aspects in agent-based systems, which define the metamodel for the SAOA: social structures and social protocols.

Social structures

Social structures represent the agent system's architecture (Shen et al. 2001) characterizing the blueprint of relationships between agents and giving a high level view of how groups solve problems, as well as the role each agent plays within the structure. There are diverse types of social structures, such as hierarchical, federated and autonomous ones. In order to understand how to define social structures, one can be inspired by the literature about classical agent frameworks for system architectures. For example, Shen et al. (2001) address social structures according to the level of centralized control implemented within the organization. Control relates to the degree of autonomy an agent possesses. In 'full control', an agent has all of its actions and goals prescribed by the imperatives of another agent. On the other hand, with 'no control' the agents involved are free to accept or reject goals, plans and actions proposed by other agents, meaning that they are free to collaborate or not. In the continuum 'full control' – 'no control', one can find several different agent organization possibilities. We classify all these possibilities into hierarchical, federated and autonomous agents' architectures (see Figure 7a).

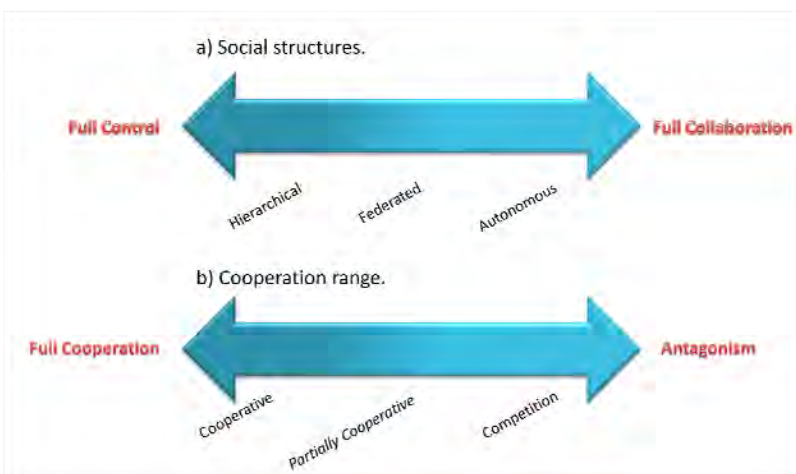

Figure 7: Social Structures and Social Protocols (inspired by Shen et al. 2001).

In hierarchical systems, a number of often distributed and semi-autonomous units exist, each with a degree of control over other local resources. In federated systems, several intermediary agents can be created to coordinate multiple agent activities, such as facilitators, brokers and mediators. Hybrid systems may also exist.

Social protocols

Social protocols refer to an agent's abilities concerning social aspects. These protocols are a set of rules governing connections between entities, defining, for example, the syntax, semantics and approaches for synchronizing interactions.

Agent's abilities concerning social aspects are normally related to cooperation (Doran et al. 1997) principles (i.e. agents have to cooperate in order to plan the entire supply chain). The concept of cooperation influences how one conceives or selects social protocols. In return, these protocols greatly influence how the system will function. In fact, agent-based systems can range from extreme cases of 'full cooperation' to 'antagonism' (see Figure 7b). In fully cooperative systems, agents are able to change their goals to suit the needs of other agents, ensuring cohesion and coordination. In this case, cooperation is necessary because no single agent has enough knowledge and resources to solve a specific problem (i.e. planning the entire supply chain), although agents might be able to solve different parts of the problem (i.e. planning each business unit individually). In contrast, in antagonist systems, agents are more competitive than cooperative. Between the two extremes, partially cooperative systems exist, and they represent the most common situation (Shen et al. 2001).

In order to allow cooperation patterns to exist, different methods can be configured in a supply chain planning system, including, for example, communication (general modus operandi, security methods, knowledge transfer approach, such as ontology and agent communication language), grouping and multiplication (agent coalitions, clusters and cloning), coordination, collaboration by sharing tasks and resources and conflict resolution through negotiation and arbitration. Coordination, collaboration and negotiation are particularly important topics in supply chain planning (e.g., Dudek and Stadtler 2005).

It is important to note that the two major issues in social agent organization (i.e. social structures and social protocols) are mutually dependent because one defines the degree of freedom of the other. Some elements of the social structure have to be reflected in the social protocols. For example, if one desires to employ a social protocol for negotiation, a total hierarchical social structure would not be recommended.

Based on these two aspects of the metamodel, one can perform requirements determination for the simulation model. Again, just as with GPA and DPA, a set of atomic requirements are gathered through interviews with the simulation stakeholders at the SAOA step. After, the atomic requirements are organized through a requirements diagram from SysML and an agent-based use case diagramming approach from AUML. An illustration is provided later in Section 4.

3.2.4 Individual Agent Organization Analysis (IAOA)

As mentioned by Ferber (1999), the task of assigning roles to every individual agent is normally regarded as the last phase in constructing an organization. The logic is that as soon as one knows what the functions to be assigned are, one defines individual specializations. These local assignments influence social protocols functioning inside their respective social structures. In addition, it also influences the local performance of the supply chain planning entities. This is the main idea of the IAOA.

At the individual level, agents can be organized according to different internal architectures. Despite the fact that there is little consensus on how to conceive the internal architectures of agents (Sanya and Hongwei 2003), diverse research works propose different ways to organize them. Shen et al. (2001) explain that agents' internal architectures are normally modular or layered, but other types exist (e.g. subsumptions). In the domain of agent-based manufacturing systems, these architectures define, for example, agent's local behaviours, local planning and local knowledge (Parunak 1998, Forget et al. 2008), plus an interaction management module (Michel et al. 2003).

In order to cope with this, the metamodel for the IAOA proposes that whatever the state of mind of an agent is (cognitive, reactive or hybrid), and whatever the internal architecture an agent employs, an agent can be described simply according to its abilities. This is the central point when performing simulation. Ability can be defined as the quality of being able to perform an action or facilitate its accomplishment. These abilities allow for the implementation of actions and the determination of the system's behaviour, as well as the determination of its related performance. Based on this notion, the metamodel defines two elements:

- *The Response Space*: stands for a collection of general abilities available for the agents, including very simple reactive abilities or sophisticated cognitive ones. For example, one agent can have a simple ability to monitor the inventory levels of the supply chain, or a complex ability to perform production planning employing an optimization method. This collection of activities includes mechanisms related to planning and scheduling, control, problem solving, perception, learning, knowledge management, interfaces, moving, anticipation, for example.

- *Capacity to Produce an Adapted Response*: represents the aptitude to choose which abilities have to be transformed into actions at a given time to respond to a specific situation. This capacity can vary from elementary to complex. The simplest possible capacity is related to a reactive 'if-then' mechanism, where no cognition is necessary. For example, if the inventory level drops to a given threshold, the agent uses its procurement ability to start a procurement action. As the agent becomes more intelligent, more complex responses can be made for some different situations. For example, the linear 'if-then' logic can be substituted by more complex approaches based on action optimization and learning.

To facilitate the description of the IAOA representing an agent's abilities, we employ a simple mathematical formulation. The basic principle of an agent-based simulation is explained by what Michel et al. (2003) define as a dynamic function. First, let us postulate that ∑ defines all possible states (e.g., inventory level) of a supply chain system. This system is based on the assumption that the environment evolution from time t to the next $t + dt$ results from the composition of actions A_t produced by an agent and also from the environment action E_t produced by its natural evolution at time t. Actions A_t and E_t are defined as $A_t = A_t^1 + A_t^2 + ... + A_t^n$ and $E_t = E_t^1 + E_t^2 + ... + E_t^n$.

The simulation problem consists in defining a time function, called *Dynamic* *D*: $\sum \mapsto \sum$, which results in the following:

$$\sigma_{t+dt} = D(\uplus(A_t, E_t), \sigma_t) \qquad (1)$$

In (1), $\uplus$ denotes the action composition operation, i.e. how action A_t produced at time t must be composed with E_t to calculate its consequence on the previous world state σ_t. This formulation is a way to simplify the conceptualization of such an operation, clarifying the complexity of the concepts hidden behind the word 'action' (including movements, decision-making, environment modification, and so forth).

Anchored in equation (1) and in the concept of abilities previously described, we introduce two new elements in (2): i) R_t : response space at time t composed of $R_t^1, R_t^2, ..., R_t^n$; and ii) C_t : capacity to produce an adapted response to a given situation at time t.

$$\sigma_{t+dt} = D(\uplus(C_t, R_t^S, E_t), \sigma_t) \qquad (2)$$

In (2), the R_t^S represents the S selected abilities from the response space R_t to be transformed into actions at time t. This selection is performed by C_t. Thus, the future world state σ_{t+dt} is a result of the composition of C_t, R_t^S and E_t on the present world state σ_t. The formulation in (2) highlights the complexity of the combinations of the two types of agents' abilities to dynamically change the system status.

These two elements of the IAOA metamodel (the response space and the capacity to select the best ability) provide guidelines to generate a set of atomic requirements through interviews with simulation's stakeholders, such as in the previous FAMASS approaches. After doing so, requirements structuring needs to be performed. Identically to the SAOA, the IAOA employs a use case diagramming approach from Heinze et al. (2000). In this case, <<*agents*>> are seen as <<*actors*>>, over whom the analysts have control. A proof-of-concept case is presented in Section 4 to illustrate the FAMASS applicability.

3.3 Relationships between the four modelling approaches

Basically, the first two analysis modelling approaches (i.e. General Problem Analysis – GPA and Distributed Planning Analysis – DPA) are sequential, that is, the GPA guides simulation analysts to define the simulation problem first and then, at the DPA, they translate the simulation problem into a set of experimentation requirements according to a distributed perspective of the supply chain planning system. In some situations, the distributed planning perspective forces analysts to change or adapt the simulation problem, thus analysts will need to revisit the GPA to revaluate the simulation problem. In addition, the transition from DPA to Social Agent Organization Analysis (SAOA) is naturally and directly done due to the fact that, at the social level, agents are mainly created based on the distributed planning entities.

Different from the GPA and DPA, the SAOA and IAOA can be done in parallel for some cases. For example, social protocols may generate internal agent abilities. The IAOA can also be directly related to the DPA level. For example, a new planning functionality at the DPA would require new individual agent abilities be introduced at this level. On the other hand, if a new algorithm merging two operational planning approaches from two different agents is introduced at the IAOA level, this will require that the DPA level be adjusted. This discussion justifies why there are arrows forming a triangle linking Distributed Planning Analysis (DPA), Social Agent Organization Analysis (SAOA) and Individual Agent Organization Analysis (IAOA) in Figure 2.

4 Proof-of-Concept Application

In order to illustrate how the FAMASS approach can support analysts in defining experimental requirements, this section presents a proof-of-concept application based on a real-scale industrial case done in collaboration with the Canadian softwood lumber industry.

4.1 Contextualization

The case represents a context where 'researchers' desire to use simulation to study some policies of the 'planning and control system' of a softwood lumber supply chain. In order to do so, the ‘researchers’ contact a ‘FAMASS analyst’ to support the modelling process.

By employing the FAMASS approach, the analyst explains what kinds of problems are possible to solve using agent technology. This helps the researchers in clarifying the problem, and allows the analyst and the researchers to speak the same language. Together, they discuss the simulation problem to produce analysis models. These analysis models will later allow the implementation of a set of simulation scenarios in an agent-based system, using an appropriate dataset.

4.2 General Problem Analysis (GPA)

4.2.1 Requirements Determination

By employing GPA guidelines, the simulation problem is defined. Several atomic requirements are identified through interviews with simulation researchers. They are:

GI. General Issues

SO. Simulation Object

- SO1. Policies: different supply chain decision policies related to planning horizon length, control level and planning method.

SE. Simulation Environment

- SE1. Sawmill: a planning and control system for a sawmill complex comprising normal business entities for a typical Canadian softwood lumber industry. As will be discussed later, this will limit the scope of the simulation system.

SQ. Simulation Questions

- SQ1. Contribution: do the control level, planning horizon and planning method really contribute to supply chain performance?
- SQ2. Interaction: does one policy influence the others, i.e. do they interact?

- SQ3. Optimum: what are the optimum planning horizons, planning methods and control levels to be implemented to minimize the impact of uncertainties in this supply chain? One has to consider classic uncertainties from the supply chain.

O. Objectives

- O1. Sensitivity analysis: the researchers desire to study which policies are most influential on performance metrics and how influential they are.
- O2. Optimization: they would like to determine which combination of particular decision policies best achieves desired performance metrics.

EI. Experimental Issues

F. Factors

- F1. Control level: represents the frequency at which one updates information about inventory levels, supply quantities from vendors and demand requests from clients. Different control levels need to be tested.
- F2. Planning horizon: represents the length of time that will be considered when preparing a plan.
- F3. Planning method: stands for the approach (or algorithm) used to produce a plan. In this experiment, two algorithms have to be employed ('forward planning' and 'urgency-directed forward planning').

U. Uncertainties

The researchers want to perform the experiments under typical supply chain uncertainties. Inspired by Davis (1993), three uncertainties are selected:

- U1. Supply: uncertainties from vendors are generated, such as supply delays.
- U2. Production: uncertainties from the production operations are generated, such as machine breakdowns.

- U3. Demand: uncertainties from customers are generated, such as demand oscillation.

K. Key performance indicators

- K1. Inventory: the simulation stakeholders decided that one has to consider the company's points of view. To do so, they selected the 'daily average planned inventory' as a KPI because it is directly related to supply chain costs, and it represents one of the main company's concerns.
- K2. Backorder: the researchers decided that one has to consider the customer's points of view. To do so, 'backorders' is selected because this KPI is directly related to the client's interest.

4.2.2 Requirements Structuring

Based on interviews with the researchers, 15 atomic requirements were gathered during the GPA phase. Requirements structuring helps organize all requirements by allowing visualization and quick identification of the relationship among them. Figure 8 presents a diagram for organizing them.

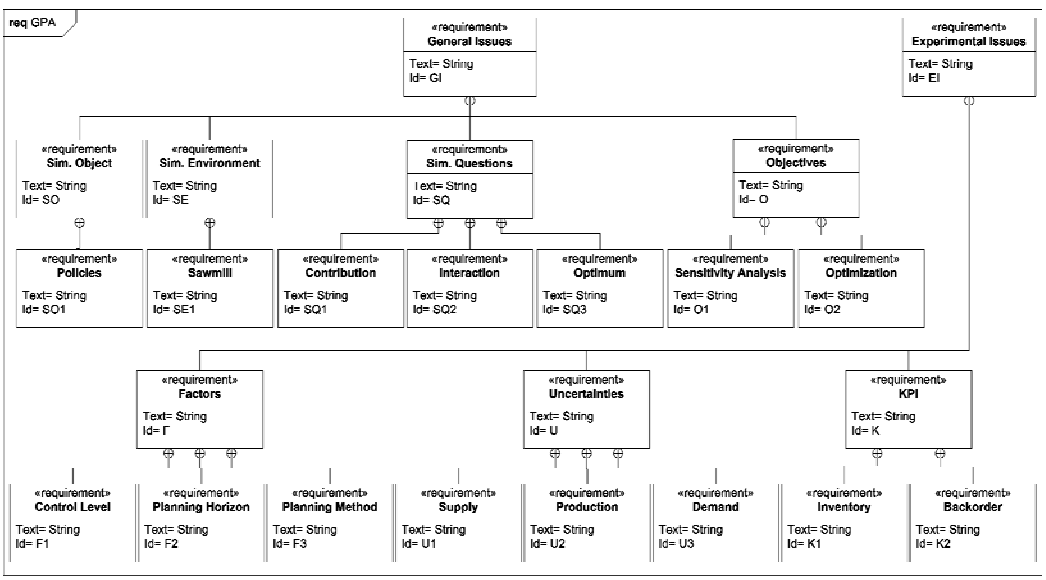

Figure 8: Requirements structuring for the GPA using a SysML requirements diagram.

Requirements structuring is particularly useful when the quantity of atomic requirements is great.

4.3 Distributed Problem Analysis (DPA)

4.3.1 Requirements Determination

Based on the proposed metamodel for DPA (i.e. the supply chain cube in Figure 3) the simulation team now has to identify the domain requirements.

Spatial dimension

- SD1. Facilities: The supply chain should represent the 'facilities' of an internal supply chain for a 'sawing complex'.

Intertemporal dimension

- ID1. Operational: the simulation experiment should comprise 'operational' entities.
- ID2. Execution: the simulation experiment should comprise 'execution' entities.

Functional dimension

- FD1. Manufacturing: the facilities within the sawing complex have to be of a manufacturing nature. No 'distribution' facilities are needed.
- FD2. Procurement: the manufacturing units have to be supported by a procurement function to manage supply. This function is at the operational level only, i.e. there is no 'procurement execution'.
- FD3. Sales: the manufacturing units have to be supported by a sales function to manage clients' demand. This function is at the operational level only, i.e. there is no 'sales execution'.
- FD4. Decomposition: using the 'decomposition approach' for a manufacturing unit (Montreuil et al. 1995) that allows entities split-up based on specialization, a

sawing complex is decomposed into three facilities: a ‘sawing’ unit, a ‘drying’ unit and a ‘finishing’ unit.

The discussion above produced seven additional atomic requirements. Before structuring these requirements, as proposed by the FAMASS approach, Figure 10 presents how the supply chain planning cube supported the simulation analysts.

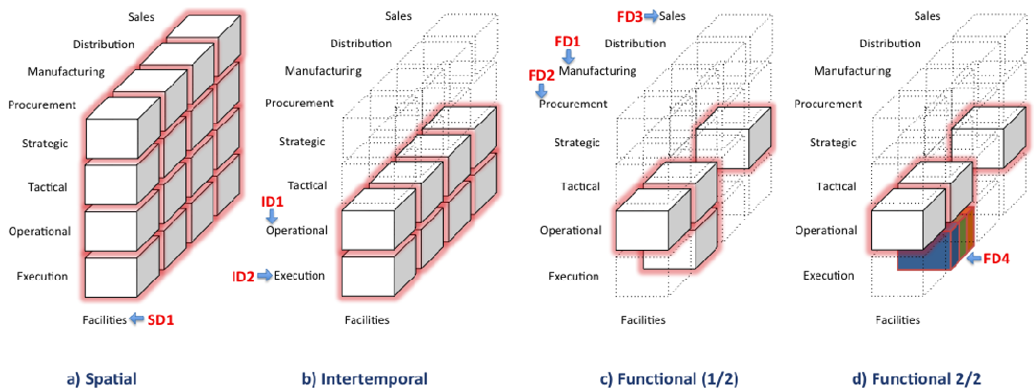

Figure 9: The selected Supply Chain Blocks for the DPA requirements.

As indicated in Figure 10, the supply chain cube supported the analysts in clarifying the scope of the distributed planning problem. All the DPA requirements are indicated in this figure. First, in Figure 10a the analysts visualize and select a spatial dimension for their experiments, i.e. the ‘facilities’ slice. At this point, they do not know which intertemporal and functional dimensions have to be examined, so the entire ‘facilities’ slice is selected. Next, in Figure 10b the ‘intertemporal’ dimension indicates that the ‘strategic’ and ‘tactical’ levels are not necessary (transparent Supply Chain Blocks). In Figure 10c shows that the ‘distribution’ and three ‘execution’ blocks are not needed. Finally, Figure 10d demonstrates the ‘decomposition’ requirement (FD4) that says that a Supply Chain Block has to be specialized into three areas, i.e. sawing, drying and finishing (the three coloured slices of the block). Thus, from the entire supply chain cube, only six distributed entities are selected. The cube helped the analysts to clarify the planning and control problem. Based on these discussions, all DPA requirements have to be structured.

4.3.2 Requirements Structuring

The FAMASS approach proposes to generate a use case diagram, which is, by definition, the 'functionality' (or the 'use') of a system from the user's perspective. Figure 10 presents the use case diagram and how it satisfies the domain requirements. It also indicates how the experimental requirements from GPA are related to the use cases of the Supply Chain Cube.

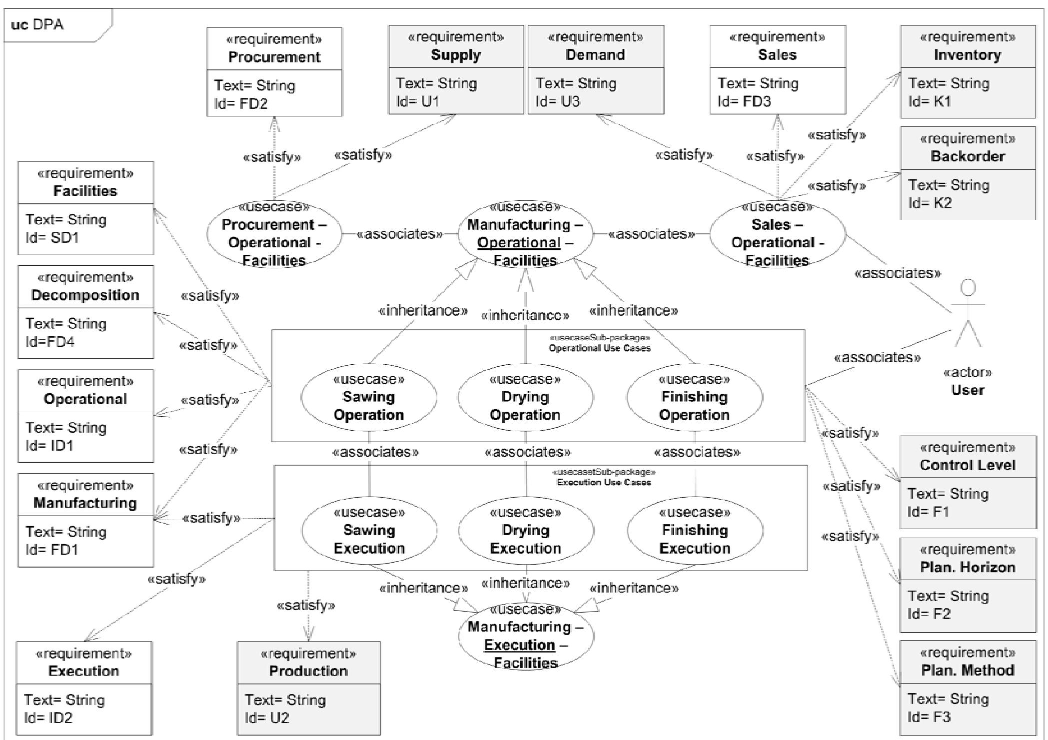

Figure 10: DPA domain requirements using a SysML use case diagram.

Note that in Figure 10, specific use cases generated at the DPA phase are in conformity with the selected atomic requirements derived from the supply chain planning cube as a reference metamodel. For example, the 'Supply Chain Block' 'manufacturing – operational – facilities' gave rise to three specialized entities (sawing, drying and finishing), by respecting a specific domain requirement, i.e. the 'Decomposition – FD4'. It is also possible to see that experimental requirements from the GPA (in grey) are still valid at the DPA. For instance, the three control factors' requirements (control level, planning horizon and planning method) are satisfied by the three 'operational' use cases (Sawing Operation, Drying Operation and Finishing Operation). The <<actor>> 'user' interacts with the operational use cases to configure simulation factors, as well as with the sales use case to verify the gathered KPI.

Based on the GPA and on the DPA, the next two FAMASS approaches transform the Supply Chain Blocks into an agent society.

4.4 Social Agent Organization Analysis (SAOA)

4.4.1 Requirements Determination

In what follows, domain requirements are created based on metamodels for the SAOA, i.e. Social Structures and Social Protocols.

SS. Social Structures

- SS1. One structure: the agent society has only one stable social structure.
- SS2. Encapsulation: all entities (use cases) of operational and executional nature are encapsulated as one individual agent each.
- SS3. Intermediary agents: there are no intermediary agents to support the other agents.
- SS4. Operational relations: all entities (use cases) of an operational nature interact with their immediate partners at the operational level (e.g. sawing operation ↔ drying operation ↔ finishing operation).
- SS5. Execution relations: all entities (use cases) of an executional nature interact only with their counterpart of the operational level (e.g. sawing execution ↔ sawing operation).

SP. Social Protocols

- SP1. New requirements: all agents of an operational nature can communicate with their corresponding supplier through a 'new requirement' protocol.
- SP2. New supply: all agents of an operational nature can communicate with their corresponding clients through a 'new supply' protocol.
- SP3. Operational plan: all agents of an operational nature can communicate with their execution counterpart through an 'operational plan' protocol.
- SP4. Execution status: all agents of an executional nature can communicate with their operational counterpart through an 'execution status' protocol.

4.4.2 Requirements Structuring

The SAOA generated nine additional atomic requirements. Based on these requirements, an agent-based use case diagram (Heinze et al., 2000) is produced (Figure 11) representing the social structure and protocols. In this case, eight <<*agents*>> are modelled as internal <<*actors*>>, confined inside the system boundaries, and their relationships are modelled as <<*associates*>>. Each agent is derived from the Supply Chain Blocks of the DPA; for example the 'Sawing Operation' block in Figure 10 creates a 'Sawing Operation' agent in Figure 11; the 'Sales – Operational – Facilities' becomes a 'Deliver' agent; and the 'Procurement – Operational – Facilities' is converted into a 'Source' agent.

After this, the SAOA recommends that the simulation team evaluate whether certain agents are to be excluded or new agents introduced, such as a mediator. In this case, a one-to-one approach for agentification was employed because no agent specialization or intermediary agents were necessary (see requirement SS3). The sole adaptation made by the simulation team was the introduction of a 'User' agent as an <<*actor*>> to represent the external environment. Four use cases were created to indicate social protocols, since these protocols can be seen as functionalities of the system. As there is no functionality related to the social structures, only agents, actors and relationships are used to represent the social structure selected by the simulation team.

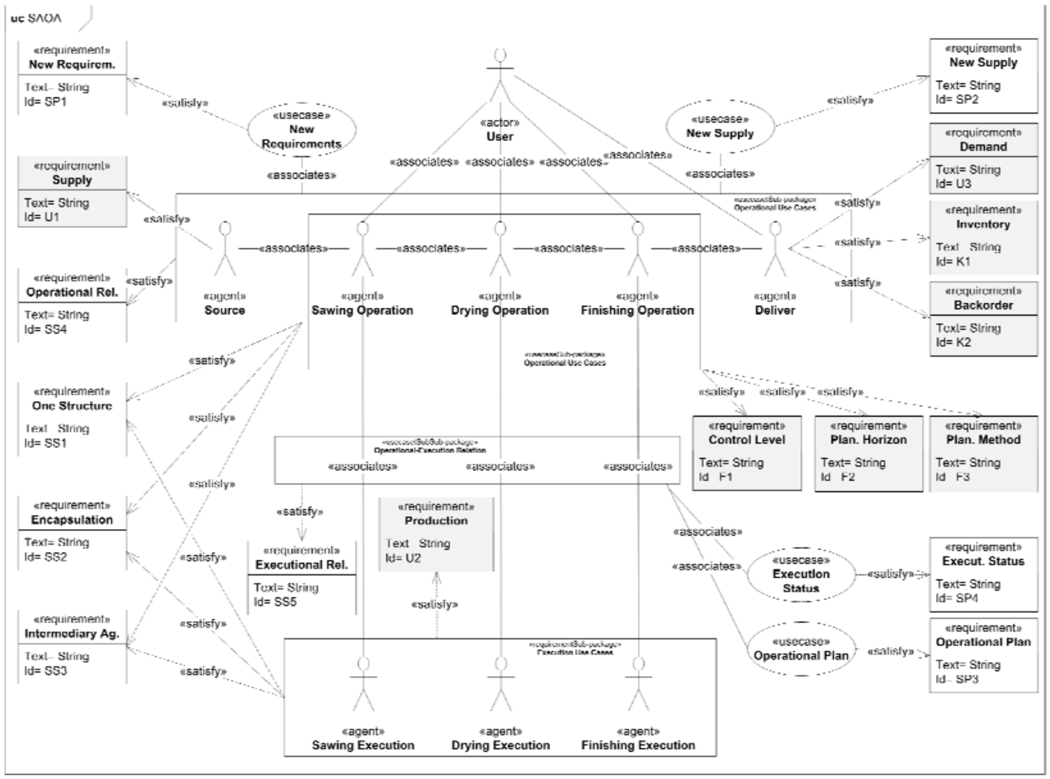

Figure 11: SAOA's social structure and social protocol using a SysML use case diagram.

Similarly to DPA, Figure 11 also indicates how experimental requirements from the GPA (the grey ones) are satisfied at the SAOA.

4.5 Individual Agent Organization Analysis (IAOA)

4.5.1 Requirements Determination

The IAOA metamodel suggests two analyses efforts, as explained next:

CP. Capacity to Produce an Adapted Response

- CP1. Planning Method Selection: agents of a 'manufacturing – operations' nature (i.e. sawing, drying and finishing) have two planning methods each, but they cannot select their planning method alone. The users are the ones who decide which planning method should be employed.

GA. General Abilities (Response Space)

- GA1. Planning & Control: agents of a ‘manufacturing – operations’ nature have planning and control abilities.
- GA2. Forward Planning: agents of a ‘manufacturing – operations’ nature are capable of using a specialized algorithm for ‘Forward Planning’.
- GA3. Urgency-Directed Forward Planning: agents of a ‘manufacturing – operations’ nature are able to use a specialized algorithm for ‘Urgency-Directed Forward Planning’.
- GA4. Short Planning Horizon: agents of a ‘manufacturing – operations’ nature are able to plan using a short time frame.
- GA5. Long Planning Horizon: agents of a ‘manufacturing – operations’ nature are capable of planning using a long time frame.
- GA6. Tight Control Level: agents of an ‘execution’ nature are able to use a tight control level.
- GA7. Loose Control Level: agents of an ‘execution’ nature are capable of using a loose control level.
- GA8. Production: agents of an ‘Execution’ nature are able to perform manufacturing activities.
- GA9. Supply: the ‘Source’ agent can generate supplies for the sawmill.
- GA10. Demand: the ‘Deliver’ agent can manage demands from clients and send them to the sawmill.
- GA11. KPI Calculation: the ‘Deliver’ agent can calculate both inventory and backorder KPI.

4.5.2 Requirements Structuring

At the IAOA level, one has to proceed exactly as at the SAOA level. For the sake of simplicity, Figure 12 presents only the use cases related to the manufacturing planning activities (i.e. sawing, drying and finishing) together with their related domain requirements (the white ones) and with the GPA experimental requirements (the grey ones).

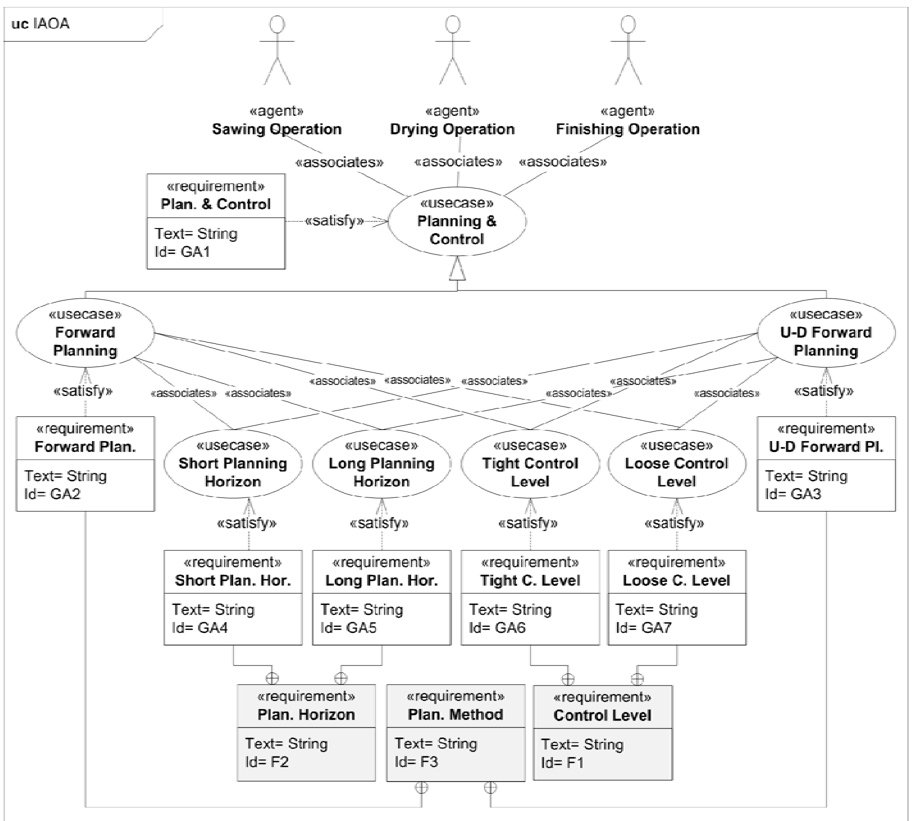

Figure 12: Partial IAOA agents abilities for the Planning & Control.

Interestingly, all experimental requirements of Figure 11 (the grey ones) related to planning issues give rise (through <<*requirementContainement*>>) to a set of domain requirements (the white ones), which in turn become directly attached to use cases. This means that these experimental requirements now possess specific system's functionalities (use cases) insuring their mission. For example, the experimental requirement F3 influenced the creation of the GA2 and GA3. These two new requirements created two new specialized use cases responsible for two different planning algorithms, the ‘Forward Planning’ and the ‘Urgency-Directed Forward Planning’. These two new use cases were derived (through <<*inheritance*>>) from the ‘Planning & Control’ use case, which is a general functionality of a supply chain planning entity. It is interesting to note that this situation did not happen at the SAOA for the present case study, because all experimental requirements were attached directly to agents (and not to use cases) at the SAOA. In consequence, they did not have any specific functionality to satisfy them. On the other hand, all experimental requirements now have dedicated functionalities, insuring the simulation team that its experiments can be configured as desired.

4.6 Analysis Deployment and Implementation

Based on the proof-of-concept case described here, a set of specification and design models were generated. Despite the fact that FAMASS only covers the analysis phase, its analysis model can be easily translated into specification and design models using an existing methodology. For the present proof-of-concept case, specification and design models were generated in accordance with the framework of Labarthe et al. (2007), a well-cited development in the field of methodological agent-oriented framework for supply chain management simulation. By doing so, it was possible to evaluate whether FAMASS was flexible and generic, and whether it could be combined with other approaches. In addition, it removed the research effort needed to develop a totally new specification and design methodology for the domain, although this could be suitable and desirable for future research initiatives.

These models were then implemented using an industrial dataset from two Canadian softwood lumber companies. Simulations were executed and data was collected and analyzed statistically. The simulation results are published in Santa-

Eulalia et al. (2011). The present paper goes further by complementing the previous publication and explaining how the FAMASS methodology was employed in this proof-of-concept case.

Simulation results indicate that supply chain control levels play a relevant role in defining robust service levels, while the planning horizon and the planning method have a lower impact in this context. In addition, from the inventory level viewpoint, it was verified that the three investigated tactical rules (control level, planning method and planning horizon) have to be configured together if one desires to maximize their contribution for a robust supply chain system capable of coping with uncertainties from the business environment. When these rules are evaluated individually, it is not possible to make the most of their potential due to interactions between them. Santa-Eulalia et al. (2011) conclude by proposing an optimum robust configuration of the tactical rules to minimize the impact of supply chain uncertainties.

5 Final Remarks and Conclusions

FAMASS is founded on a set of complex and different concepts from many disciplines. The interest of the proposed analysis methodology relies on the simplicity of its application, as identified during the proof-of-concept case. It proposes to analyze simple elements and organize them according to well-established formalisms, providing a straightforward application useful for any simulation analyst.

The fundamental contribution of the proposed framework lies in the semantic unification mechanism it provides for simulation stakeholders (users, modellers, analysts, developers, etc.) to deal with system complexity. This mechanism not only supports knowledge capturing and retention, but it also facilitates knowledge sharing through the use of a common ‘language’ by all stakeholders.

In addition, the proposed framework provides a way to eliminate a typical problem in agent-based systems, i.e. the engineering divergence phenomenon (Michel et al. 2003). This occurs when the conceptual model is incomplete at the analysis phase, causing the system to be configured or implemented in different ways, consequently yielding outputs that are different from the simulation stakeholders' real requirements. In contrast, FAMASS harmonizes simulation requirements to fulfil stakeholders' needs, increasing the modelling process's convergence.

Several future developments are being envisaged for FAMASS, including the incorporation of advanced requirements structuring approaches, the development of a complete deployment strategy towards specification and design, and the incorporation of infrastructure issues. Future versions of FAMASS are to be published in the near future.

Acknowledgments

The authors wish to thank the NSERC (National Science and Engineering Research Council of Canada) and the FORAC Research Consortium (www.forac.ulaval.ca) for their financial support, as well as the two anonymous referees for their valuable recommendations to improve the manuscript.

References

Baumgaertel, H. and John, U., 2003. Combining agent-based supply net simulation and constraint technology for highly efficient simulation of supply networks using APS systems. *In*: *2003 Winter Simulation Conference,* 7-10 December 2003, New Orleans.

Chatfield, D.C., Hayya, J.C., and Harrison, T.P., 2007. A multi-formalism architecture for agent-based, order-centric supply chain simulation. *Simulation Modelling Practice and Theory*, 15(2007), 153–174.

Chu, H., Boonserm, K., Jeong, H. and Jones, A., 2004. Using business process specifications and agents to integrate a scenario-driven supply chain. *International Journal of Computer Integrated Manufacturing*, 17(6), 546-560.

Davis, T. 1993. Effective supply chain management. *Sloan Management Review*, 3(4), 35-46.

Derrouiche, R., Neubert, G. and Bouras, A., 2008. Supply chain management: a framework to characterize the collaborative strategies. *International Journal of Computer Integrated Manufacturing*, 21(4), 426-439.

Doran, J.E., Franklin, S. Jennings, N.R. and Norman, T.J., 1997. On cooperation in multi-agent systems. *The Knowledge Engineering Review*, 12(3), 309-314.

Dudek, G. and Stadtler, H., 2005. Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research*, 163 (2005), 668–687.

Egri, P. and Vancza, J., 2005. Cooperative planning in the supply network – a multiagent organization model. *In*: *CEEMAS 2005 - 4th International Central and Eastern European Conference on Multi-Agent Systems*, 15-17 September 2005, Budapest.

Ferber, J., 1999. *Multi-agent systems: an introduction to distributed artificial intelligence*. Harlow: Addison-Wesley.

Fontanili, F., Castagna, P. and Yannou, B., 2008. Les outils de simulation. *In*: C. Thierry, Thomas, A., and Bel, G. *La simulation pour la gestion des chaînes logistiques*. Paris: Lavoisier.

Forget, P., D'Amours, S., and Frayret, J.-M., 2008. Multi-behaviour agent model for planning in supply chains: an application to the lumber industry. *Robotics and Computer-Integrated Manufacturing Journal*, 24(5), 664-679.

Frayret, J.-M., S. D'Amours, A. Rousseau, S. Harvey and J. Gaudreault, 2008. Agent-based supply-chain planning in the forest products industry. *International Journal of Flexible Manufacturing Systems*, 19(4), 358-391.

Galland, S., Grimaud, F., Beaune, P. and Campagne, J.P., 2003. MAMA-S: an introduction to a methodological approach for the simulation of distributed industrial systems. *International Journal of Production Economics*, 85, 11–31.

Govindu, R. and Chinnam, R., 2006. Methodological frameworks for multi-agent supply chain modeling. *In: SIAM Symposium on Complexity and Business Analytics: Theory and Applications*, 14 October, Detroit.

Heinze, C., Papasimeon, M. and Goss, S., 2000. Specifying agent behaviour with use cases. *In*: Z. Chengqui and S. Von-Wun, eds. *Lecture Notes In Computer Science: Design and Applications of Intelligent Agents*. Berlin: Springer-Verlag, 1881(2000), 128-142.

Hoffer, J., George, J. and Valacich, J., 2005. *Modern systems analysis and design*. New Jersey: Pearson Prentice Hall.

Huget, M., 2004. Agent UML notation for multiagent system design. *IEEE Internet Computing*, 8(4), 63-71.

Ivanov, D., 2009. DIMA—A research methodology for comprehensive multi-disciplinary modeling of production and logistics networks. *International Journal of Production Research*, 47(5), 1153–1173.

Karam, M., Tranvouez, E., Espinasse, B. and Ferrarini, A., 2010. An organization-oriented methodological framework for agent-based supply chain simulation. *In:2010 Fourth International Conference on Research Challenges in Information Science (RCIS)*, 19-21 May 2010 Nice.

Labarthe, O., Espinasse, B., Ferrarini, A. and Montreuil, B., 2007. Toward a methodological framework for agent-based modelling and simulation of supply chain in a mass customization context. *Simulation Modelling Practice and Theory*, 15(2), 113-136.

Lapide, L., 2000. What about measuring supply chain performance? *AMR Research*

White Paper - Achieving Supply Chain Excellence Through Technology, ASCET, 3(2000), 287-297.

Lee, J. and Kim, C., 2008. Multi-agent systems applications in manufacturing systems and supply chain management: a review paper. *International Journal of Production Research*, 46(1), 233-265.

Meyr, H. and Stadtler, H., 2004. Types of supply chain. *In*: H. Stadtler and C. Kilger, eds. *Supply chain management and advanced planning: concepts, models, software and case studies*. Berlin: Springer.

Michel, F., Gouaïch, A. and Ferber, J., 2003. Weak interaction and strong interaction in agent based simulations. *Lecture Notes in Computer Science*, 2927(2003), 43-56.

Monteiro, T., Anciaux, D., D'Amours, S., Espinasse, B., Ferrarini, A., Labarthe, O., Thierry, C. and Thomas, A., 2008. L'intérêt des agents pour la simulation de la chaîne logistique. *In*: C. Thierry, A. Thomas, and G. Bel, eds. *La simulation pour la gestion des chaînes logistique.* Paris: Lavoisier.

Montreuil, B. and Lefrançois, P., 1996. Organizing factories as responsibility networks. *In*: R.J. Graves, L.F. McGinnis, D.J. Medeiros, R.E. Ward and M.R. Wilhelm, eds. *Progress in Material Handling Research*. Michigan: Braum-Brumfield, 375-411.

Montreuil, B., Boctor, F., and Martel, A., 1995. La maîtrise des technologies de production. *In*: A. Martel, and Oral, M. eds. *Les défis de la compétitivité : vision et stratégies*. Montréal: Publi-Rélais.

OMG, 2010. OMG Systems Modeling Language (OMG SysML™). *Object Management Group Specification Report*, June 2010.

Parunak, H., 1998. Practical and industrial applications of agent-based systems [online]. Industrial Technology Institute, Available from: http://agents.umbc.edu/ [Accessed November 2007].

Robinson, S., 2004. *Simulation: the practice of model development and use*. John Wiley and Sons.

Robinson, S., 2008. Conceptual modeling: definitions, concepts and future research. *In: Workshop on Conceptual Modeling – Spring Simulation Multiconference*. Ottawa.

Sauter, J. A., Parunak, H. V. D., and Goic, J., 1999. ANTS in the supply chain. *In:Agents'99 Workshop Agent-based decision-support for managing the Interned-enabled supply-chain*. Seattle.

Santa-Eulalia, L.A., Ait-Kadi, D., D'Amours, S., Frayret, J.-M. and Lemieux, S., 2011. Agent-based experimental investigations of the robustness of tactical planning and control policies in a softwood lumber supply chain. *Production Planning and Control*, In press in iFirst (first published online on 10 February 2011).

Santa-Eulalia, L., 2009. *Agent-based simulations for advanced supply chain planning: a methodological framework for requirements analysis and deployment*. Thesis (Ph.D.). Université Laval, Faculté des Sciences et Génie.

Santa-Eulalia, L., D'Amours, S. and Frayret, J., 2008. Essay on conceptual modelling, analysis and illustration of agent-based simulations for distributed supply chain planning. *INFOR Information Systems and Operations Research Journal*, 46(2), 97-116.

Sanya, L. and Hongwei, W., 2003. Agent architecture for agent-based supply chain integration and coordination. *Software Engineering Notes*, 28(4).

SCOR, 2006. Supply-Chain Operations Reference-model (SCOR) - Version 8.0.

Supply-Chain Council, Inc.
Shapiro, J., 2000. *Modeling the supply chain*. Duxbury: Pacific Grove.
Shen, W., Norrie, D. and Barthès, J., 2001. *Multi-agent systems for concurrent intelligent design and manufacturing*. London: Taylor & Francis.
Stadtler, H. and Kilger, C., 2004. *Supply chain management and advanced planning: concepts, models, software and case studies*. Berlin: Springer.
Swaminathan, J., Smith, S. and Sadeh, N., 1998. Modeling supply chain dynamics: a multiagent approach. *Decision Sciences*, 29(3), 607-632.
Van der Zee, D. and Van der Vorst, J., 2005. A Modeling framework for supply chain simulation: opportunities for improved decision making. *Decision Sciences*, 36(1), 65-95.
Vernadat, F., 1996. *Enterprise modeling and integration: principles and applications*. London: Chapman & Hall.
Vernadat, F., 2002. UEML: towards a unified enterprise modelling language, *International Journal of Production Research*, 40 (17), 4309-4321.
Wooldridge, M., 1998. Agents and software engineering. *AI*IA Notizie*, 11(3), 31-37.