8th International Conference of Modeling and Simulation - MOSIM'10 - May 10-12, 2010 - Hammamet - Tunisia
"Evaluation and optimization of innovative production systems of goods and services"

AGENT-BASED SUPPLY CHAIN SIMULATION: TOWARDS AN ORGANIZATION-ORIENTED METHODOLOGICAL FRAMEWORK

Karam MUSTAPHA, Erwan TRANVOUEZ, Bernard ESPINASSE, Alain FERRARINI
LSIS UMR CNRS 6168 – Aix-Marseille University
Domaine Universitaire de Saint-Jérôme 13397 MARSEILLE CEDEX 20
{karam.mustapha, erwan.tranvouez, bernard.espinasse, alain.ferrarini}@lsis.org

ABSTRACT: *The Supply Chain (SC) organizational structure, and related management policies, is an important factor that can be adjusted to improve the SC performance, which consequently has to be taken into account in the SC modeling and simulation. In the context of an agent-based SC simulation, this paper addresses a new methodological framework, organizationally oriented, which permits modeling and simulation of such SC organizational aspects, allowing observables of different levels of details while reproducing the SC behavior according to desired observables. This methodological framework is structured according to two main abstraction levels, a conceptual level and an operational level. At the conceptual level, a Conceptual Role Organizational Model (CROM) is produced, and then refined into a Conceptual Agent Organizational Model (CAOM). At operational level, CAOM model is translated into an OPerational Agent Model (OPAM) which will be implemented and simulated in a specific agent based software architecture briefly introduced in this paper.*

KEYWORDS: *Agent Based Simulation, Multi-Agent Systems, Supply Chains, Organization.*

1 INTRODUCTION

The Supply Chain (SC) domain is a rich playground for complex studies. Simulation of such systems aims to experiment and understand (in a controlled environment) the economical, human and environmental consequences of decisions related to the organization, the management policies and the design of the production facilities. The dimensions of the problem are numerous, and the conceptual and architectural challenges, that SC dedicated simulation-based decision support systems raise, imply a heavy workload. We propose to enable the study of the efficiency of production organization and decision-making processes which supposes to: i) describe the SC organization; ii) model and simulate the behaviors and decisions of its actors and iii) implement these decisions and see their local and global effect on the SC, iv) support each step with specific conceptual and software support. Our work involves a methodological and architectural framework which assists the SC experts in producing and experimenting with distributed simulation of their SC's based on a multiagent perspective.

Multiagent or Agent Based Simulation (ABS) contribution to SC studies is established [Bruekner et al., 2003] [Shen et al., 2006] [Monteiro et al., 2008]. Agents are exploited for the design and/or the simulation of complex systems, as autonomous entities that are able to perform their functions without the need for continuous interaction from the user. Agent-Based Simulation allows focusing on the behaviors of the various SC's actors.

LSIS previous work [Labarthe et al., 2007] has proposed the basis of a methodological framework for helping SC experts to design their models in their own language (domain models), as well as transitional agent-based models which are used to produce the distributed simulation model on which experiments are conducted.

Our current work aims to take into account the impact an SC's organizational structure has on its performances by providing a methodological framework which support ranges from the domain model analysis to running the simulation. The methodological framework, in line with our former works, considers the design phase according to a conceptual level and an operational level. This framework has to facilitate the realization of the SC simulation with gradual processes that begin by defining the needs of the user prior to arriving to the implementation of the system while meeting the initial requirements. Enacting this methodological framework requires i*n fine* a software architecture adapted to the need of SC simulation. The proposed framework should take into account the objectives of different modeling paradigm, as well as heterogeneous (simulation) software environments to coexist. Thus, final users would avoid any loss of previous expertise in modeling language which they have chosen for legitimate scientific reasons and comfort in use. This paper focuses on the main models of this methodological framework and introduces the general software architecture.

Firstly, we define our research problem in section 2, which includes modeling and simulation of SC organizational aspects. In section 3, we introduce the methodological framework allowing to take into account SC organizational aspects. Then in section 4, we detail the conceptual organizational modeling stage of this methodological framework, with the *Conceptual Role Organizational Model* (CROM), which is refined into a *Conceptual Agent Organizational Model* (CAOM). Section 5 describes the operational organizational modeling stage of the methodological framework, centered on the *OP-*

erational Agent Model (OPAM) obtained by translation from the CAOM. Finally, we conclude by drawing the future step of our research concerning the specification of the software architecture for OPAM implementation and simulation.

2 SC ORGANIZATIONAL ASPECTS AND THEIR MODELING AND SIMULATION

Based on an agent oriented approach, our research aims to propose a methodological framework that takes into account the impact that an SC's organizational structure has on its performances, from the domain model analysis to running the simulation. In this section, we first try to define the problem of modeling the SC organizational aspect, then we present the requirements of simulation of such organizational aspects, and finally we develop this problem in an Agent Based Simulation context.

2.1 Modeling SC Organizational Aspects

The objectives of the work are to propose a SC simulation methodological approach which begins by identifying the observables, describing a SC organization and then designing the simulation model. In the field of modeling methodology for SC simulation, few approaches take explicitly into account an organizational point of view and user defined observables. Observables are data and information related to ongoing logistic processes, which need to be highlighted in the simulation results for a particular study. Observables can be simple or aggregated (at different hierarchical levels) informations describing the states of the SC entities, performance indicators as well as operational processes states or decision processes assessments (scheduling strategies, stock management strategies, etc.) and their consequences (performance evaluation of their outcomes on the SC). Most approaches focus either on management process modeling (such as SCOR "Supply Chain Operations Reference model" [SCC, 2006]), or on a static point of view of the hierarchical description of the enterprises involved in the chain. Therefore, the main goal of this issue is to reproduce the SC behavior according to the level of details required to produce the user desired observables.

LSIS researches [Labarthe et al., 2007] proposed a first modeling framework focusing on interrelation dynamics, which did not explicitly define the hierarchical organization. Moreover, the underlying modeling analysis was guided by the nature of the logistic processes (duality physical / decisional) and thus constrained the representation of the organization at the simulation level. However, the organization is the medium in which observables evolve as they are produced, aggregated, transformed along the organizational actors, and or hierarchical levels. The need for an approach which considers this organizational/observables issue then appears.

In this perspective, the organization is seen as representation of role distribution between actors and by determination of relationships between these actors. On one side, the SC organization must be analyzed and described in the design phase: identification of actors and the organizational structure to which they belong, modeling of their behaviors (Domain and Conceptual Models in fig. 1). On the other side, an operationalization phase should emphasize their behaviors (ie in operational models and implementation). Such an approach also needs to explicitly take into account the observables the users want to focus on during simulation.

All the observables are assumed to be modeled as indicators. An indicator is usually defined as selected information associated with a phenomenon, designed to observe periodic changes in the light of objectives. Therefore, it is a quantitative data that characterizes an evolving situation (an action or consequences of an action) in order to evaluate and compare their status at different time stages.

Therefore, a modeling and simulation method allowing exhibiting *in fine* an actor centered organization that focuses on the SC actors and the behavior of these actors, making emerge user desired observables, appears worthwhile.

2.2 Simulation of SC Organizational Aspects

In a SC, information and processes are naturally distributed in the physical organization. In order to achieve simulation of actors' behaviors according to the organization structure, distributed capabilities are thus required to fully exploit the models introduced in the previous section. Moreover, in order to highlight the desired observables the simulation architecture must deal with a set of constraints and requirements identified by [Gaud et al., 2008] [Labarthe et al., 2007] [Espinasse et al., 2007] [Vangheluwe et al., 2002].

- *Multi-level simulation*: It should be possible to describe different levels of detail in the SC infrastructure such as the organization of production cell, a transport company fleet or a company of the SC.
- *Multi-scale simulation*: Simulating all the components along different scale of the SC may not be relevant or efficient from a modeling complexity or technical point of view.
- *Multi-paradigm modeling*: It is partly related to the above requirement. Behaviors of the SC entities can be coupled and may require relatively high level of description/modeling capabilities (to reproduce / validate negotiation or planning processes or protocols) or lower level of description (simple behavior such as a simple production machine, a truck, etc.).
- *Managing different temporal scale*: The simulation must deal with local schedulers while ensuring consistent the global behavior of the SC (in terms of time constraint and causality) since the behaviors of the SC entities can either describe activities in real-time or with longer duration (e.g. a rescheduling process).
- *Openness to modeling or simulation legacy software*: This interoperability is related to the reuse of important modeling and/or learning efforts previously done. It can cover the compatibility with previous research/simulation results.

The time simulation of these models implies distributed software architecture. Two main approaches are possible:

- Define generic (homogeneous) agent based architecture (with dedicated modeling language) [Gaud et al., 2008] [Hubner et al., 2008].
- Coordinate separate simulations (particularly when different paradigms are used) through interoperability mechanisms and protocol as HLA [Ounnar et al., 2008].

The presented work aims to go towards convergence using the above two approaches, in respects with the previous modeling and simulation requirements listed earlier. First of all, it is made by using an organizational oriented and individual-based modeling approach, which is simple enough to be related to the domain-dedicated modeling language, and also by producing models which afterward can be translated into another modeling paradigm and language. Secondly, it is maintained by proposing an agent based framework that keeps different models and simulations consistent independently of the software environment in which they are implemented in (as in [Espinasse et al., 2007] in an environmental decision support context). Moreover, this software framework must be sufficiently open to other simulation software environments. This paper focuses on the first part of the problem and highlights the main outlines of the second part toward the operationalization and software architecture.

2.3 Agents based Approach and SC Organizational Aspects Modeling and Simulation

ABS allows the understanding of different dynamic models, which are composed of entities with different complexity levels (from very simple entities or reactive agents to more complex such as deliberative agents). Another interest of ABS is the facility offered to the modeler to manipulate different levels of representations, such as individuals and groups of individuals. Agent-based modeling allows capturing of the dynamic nature of SCs, facilitating the study of numerous resources coordination that is associated with the interaction of multiple companies [Monteiro et al., 2008].

Agent based SC simulation is now frequently used, but few researchers have proposed a general framework to support both the design and the realization of the SC simulation. Among those, the MASCF methodology (Multi-Agent Supply Chain Framework) [Govindu and Chinnam, 2007] adapts the SCOR model to a structured generic methodology for multi-agent system development (Gaia). However, the organization modeling is based on a management process metaphor which underrates the organizational structure. A more general study of agent oriented software engineering methodologies (among those rare holonic compliant methods), undertaken in order to find conceptual and operational solutions, confirmed that organizational issues were to be added to the actor approach [Labarthe et al., 2007]. Methods like GAIA [Zambonelli et al., 2003], CRIO [Gaud et al., 2008], Luis Antonio [Antonio et al., 2008] or MOISE+ [Hubner et al., 2007] provide only a part of the solution for the required objectives.

Almost all the previously cited approaches use roles in order to promote the flexibility of the developmental process, even with different abstraction or hierarchical levels. Roles are either used to decompose or to contextualize the behaviors of the agents and can even add constraints to them (through rules or norms). However, time is not an issue (apart for CRIO) as homogeneous agent granularity or type is mostly involved. When deliberative agents and reactive agents reproduce behaviors of different time horizon, then time synchronization becomes a hard requirement to be identified at the modeling phase and eventually controlled at the software level (and maintained at the intermediate translation steps).

As the organization of the systems pre-exists the agent model, the description of the organization must be included from the beginning of the modeling approach, in order to propose the suitable observables of its components. The group and the holon concepts respond to this requirement. Finally, cooperative behaviors are basic tools to reproduce cooperation situation in a "real" SC as well as a way to deal with disrupting events, giving its adaptability to the SC [Tranvouez and Ferrarini, 2006]. The deliberative/reactive agent architecture results directly from the need of validating such cooperative behaviors.

In conclusion, based on an agent oriented approach; we need a new methodological framework that takes into account the impact of SC's organization on its performances, from the domain model analysis to running the simulation. The following section presents such a methodological framework.

3 A METHODOLOGICAL FRAMEWORK FOR SC ORGANIZATIONAL ASPECTS MODELING

The complexity of SC modeling and simulation process as well as implementation support, lead us to propose a modeling approach based on an incremental process, relying upon models with gradual increasing details. The real system is first represented by a domain modeling of SCs (e.g. a NetMan model as in [Labarthe et al., 2007], an UEML model [1] etc.) allowing to represent the organizational aspects.

The organization-oriented methodological framework we propose is structured according to two main abstraction levels, a *conceptual level* and an *operational level*. From the domain model provided by the domain expert, a simulation model is built step by step. The conceptual level proposes concepts and models helping to grasp the complexity of the SC and its simulation objectives, whereas the operational level prepares for the implementation of the simulation model including software integration issues. The different models and the transition to

[1] Unified Enterprise Modeling Language - www.ueml.org

agent-oriented modeling and simulation in our methodological framework are presented in Figure 1.

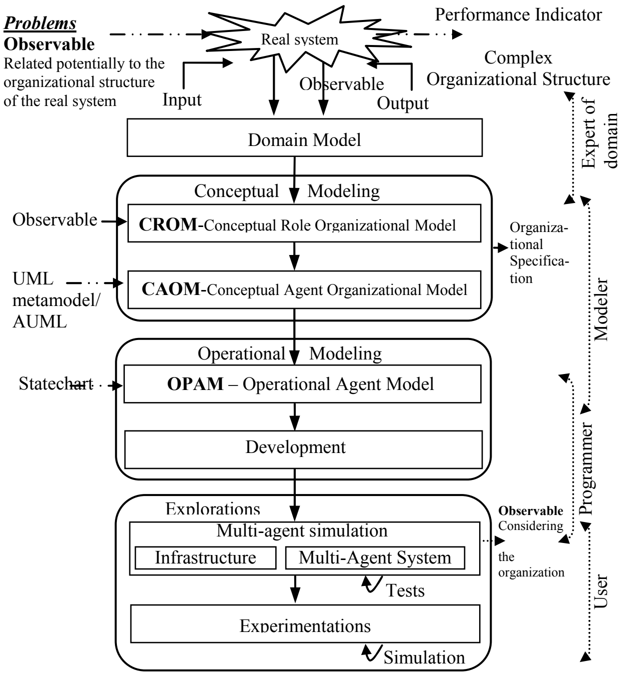

Figure 1: Methodological for the modeling and simulation oriented agents

Conceptual Organizational Modeling:

The *Conceptual Organizational Modeling* engages through a dialogue between the domain expert and an agent knowledgeable modeler. An actor model is produced by identifying the active entities and their organization from the domain model according to the role concept. The modeler has to translate/abstract the domain model into a Conceptual Organizational Modeling based on (hierarchical) levels, actors, roles and groups named *Conceptual Role Organizational Model* (CROM).

This stage highlights the organizational structure of the SC as wells as the structural and dynamic relations between the entities composing this SC. Then, a conceptual agent-based model is produced on the basis of observables which the user needs to obtain from the simulation building up the route toward the implementation of the simulation. This model is then transposed into the agent world (at a conceptual level) concluding the phase of "specification" with a multi-agent model and organization named *Conceptual Agent Organizational Model* (CAOM) ready to be described at a architectural and software design level.

The important key of this step is to precisely identify the agents defined at the conceptual level, in order to make them operational at the operational level.

Operational Organizational Modeling:

The *Operational Organizational Modeling* provides a solution to implement an executable system to perform simulations based on the previous conceptual models. This step involves the choice of agent architectures, in order to allow a differentiation between different kinds of agents. This process is guided by the observables selected earlier by the domain expert.

The software designer details the CAOM by associating a conceptual agent with a software agent architecture (for example BDI (Believe, Desire, Intention) [Rao et al. 1991]) and specifying their behaviors (for example a UML[2], state chart for a reactive agent) and interactions (AUML[3] sequence diagram [Odell et al., 2001]), resulting in an Operational Agent Model (OPAM). The implementation of these models in a simulation(s) environment results in an Agent-Based Simulation system which is then executed. This last stage of the conception requires the realization of many tests for the validation of the multi-agents system.

In [Labarthe et al., 2007] the observables, potentially related to the organizational structure of the real system, are not currently described in the design model. They are only mentioned in the multi-agent system model i.e. only one step before implementation. It is necessary to describe them to previous levels (conceptual and operational level). A second objective of our work is to propose a business model that is adequately open to different software platforms in order to facilitate the process from translation into implementation.

4 CONCEPTUAL ORGANIZATIONAL MODELING

A methodology should provide an appropriate set of abstractions to identify, develop and describe a problem and propose its potential solutions. The methodological framework presented in this paper covers the conceptual modeling area in Figure 1. The domain model (defined beforehand) has to be progressively transformed into a running simulation with heterogeneous observables. In this section, first we present the *Conceptual Role Organizational Model* (CROM) and then the *Conceptual Agent Organizational Model* (CAOM) composing this conceptual organizational modeling.

4.1 Conceptual Role Organizational Model (CROM)

This model allows highlighting the organizational structure of the SC as wells as the structural and dynamic relations between the entities composing this SC. First we present the founding concepts of this CROM model, and then its metamodel. Finally we present an illustrative example of a CROM for a specific SC.

---

[2] Unified Modelling Language;

[3] Agent Unified Modeling Language

4.1.1 Founding concepts

The founding concepts of our modeling approach are defined by a role oriented metamodel based on the existing methodologies evoked in section 3. This meta-model has to precisely define all concepts used in the development process. Our *Conceptual Role Organizational Model* or CROM meta-model extends the Agent Actor model proposed in [Labarthe et al., 2004] and adds the organizational representation capacity. We have integrated the concept of a group included in AGR that incorporates the concept of hierarchy. The recursive description of an organization hierarchy of CRIO helped us to apprehend its implication from a conceptual and architectural point of view. As an interface between the agent and the role, the notion of capacity represents an interface between two adjacent abstraction levels in the hierarchy of the system.

A system is described as a successive hierarchical layer, denoted as level, regrouping a set of roles of the same hierarchical decomposition level from the domain model. A level is in general characterized by a single time horizon (for example real time). The CROM model integrates the notions of actor, group, role, service and relation (Example fig.3):

- An *actor* is an active entity in the organization.
- A *group* represents a set of roles in the organization sharing a common goal (derived from the domain model).
- A *role* represents the functional position played by an actor in the context of a group.
- A *service* is a function performed by the role of an actor [Ferber et al., 2009].
- A *relation* is an interaction between entities.

As these concepts are used to translate SC domain models, a CROM model is associated to a domain ontology i.e. in the present case a SC ontology. Such ontology can propose a library of role hierarchy collecting the different roles a particular SC uses, as well as predefined groups type (e.g. different production service organization).

4.1.2 CROM Meta-Model

The CROM meta-model in Figure 2 shows how these concepts constitute the building blocks of a CROM model. We consider that an organization is composed of (hierarchical) levels that hold one or several groups, each group containing actors playing roles. An actor plays one role in one group, but can also play the same role in different groups. Conversely, the same role can be played by several actors. Organization, group, and actors can generate observables (quantitative or qualitative). A role provides services to other roles of the same group, while a service may require capacities (as defined in the domain model). Relations connecting actors may exist between actors and/or roles. They represent a flow of information and/or physical exchanges (products, semi finished products, raw material…) [Labarthe et al., 2007].

There are two type of groups: *structural* (it is an isomorphic description of the SC organization) or *dynamic* (i.e. characterized by a time duration or a goal shared by actors from different structural groups). Structural and functional relation sub-types relates to the same distinction. A structural group thus holds only structural relation. The different relations are detailed in Table 1.

Relation type	Categories of relation	Flow Type	Graphical notation	Description
Structural	Collaboration	information	◄··►	Represent collaborative process between actors of a same sub-structure (e.g/ cooperative scheduling). Rerquires some autonomy)
	Control	information	──►	An actor has (hierarchical) control over another i.e. he can order others to do specific tasks.
	Scheduling	information materials	····►	Organize the realization of tasksconsidering time constraints (deadlines, …)
	Planning	information	──→	Planning is the implementation of objectives over time.
Functional	Contractual	information materials	←──→	General interaction between actors outside their structural group. Relation can be limited in time. (e.g. command passing between a. customer and its supplier).

Table 1: Examples of relation types

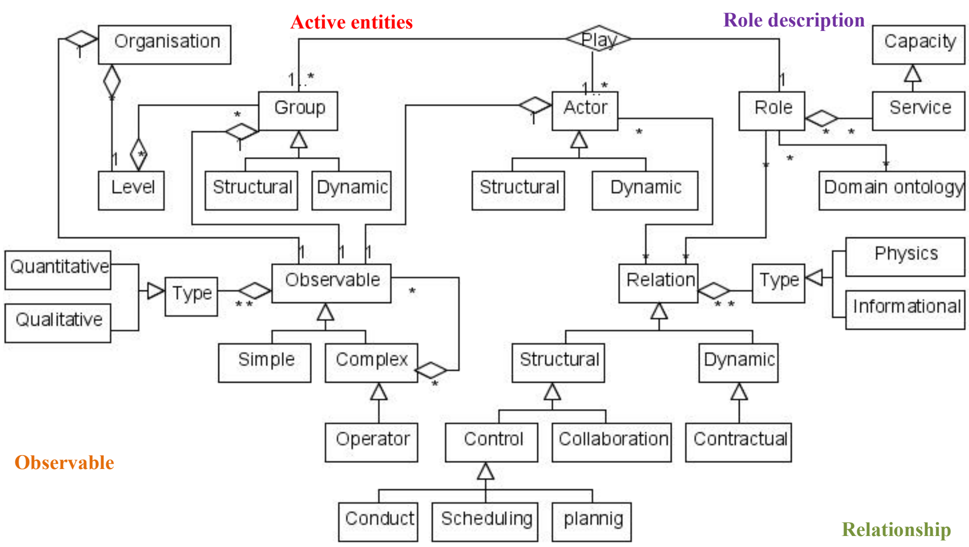

Figure 2: CROM Metamodel

The observable is characterized by the activity it monitors (productivity, quality, cost…), its quantitative or qualitative nature which requires defining its measuring units and the authorized values (whole or real number if quantity, list of values if qualitative) and finally its dated value.

As a CROM model is defined by a set of actors, groups, and hierarchical levels, a level is defined by its groups, their actors and their roles. The roles are characterized by a service that they set in motion. Thus, the structure of the group is defined like a quintuplet: $G_i=\{Ac_i,R_i,S_i,Re_i,T\}$, where Ac_i is the set of actors represented by the group G_i, R_i is the set of the roles played by Ac_i, S_i is the set of the services of the roles, Re_i is the set of the relations between the actors, and T denotes the time horizon. Each level has a temporal horizon {short term, medium term, long term} which will be of some use for the simulation.

4.1.3 An Illustrative CROM model of a SC

The following example illustrates how a CROM model is used to describe the organizational structure of a given SC. In this example, the structure is divided into 3 levels.

Each level consists of one or more groups (structural or dynamic) of actors. In the first level, *Company 1*, *Company 2* and *Company 3* are three actors connected by a collaboration relationship; each of these companies is represented by a group of actors playing roles at different hierarchical level. Actor *Company 1* for example can negotiate with actor *Company 2* products orders at the N1 level, while checking at level N2 their respective production/transport capacities with their *Production manager*/*Truck*. These last actors take short term decisions on this level but are responsible to enact these decisions and thus control in real time their execution on the third level.

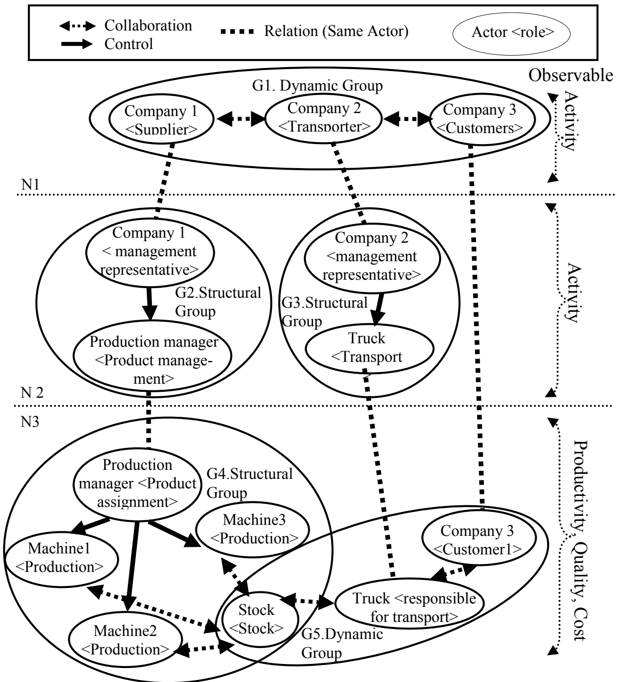

Figure 3: A CROM of the considered SC

In the given example, a VMI (Vendor Management Inventory) process is described where *Company 2* uses company 1 as a stock resource when needed. Whereas the stock actor belongs to *Company 1* (structural relationship) plays the same role <*stock*> in the dynamic group constituted with the *Truck* actor from *Company 2*. Control relationship specifies the flow of information with actors use in order to accomplish their objectives.

4.2 Conceptual Agent Organizational Model (CAOM)

The *Conceptual Agent Organizational Model* (CAOM) is a translation of the CROM in a conceptual agent modeling. Agents are not detailed as they will be described in the operational agent model according to specific agent architectures. This classical step in Agent Oriented Software Engineering associates roles with the agent according to the chosen agent modeling approach. First we present the meta-model of this CAOM (*Conceptual Agent Organizational Model*) model, and then we present an illustrative example of a CAOM model for the previous considered SC.

4.2.1 CAOM Metamodel

CAOM objective is to specify the behavior of each model actor CROM. It involves filtering the behavior of a "physical system" of the real world on the one hand, and on the other hand, the behavior of a "complex system" which processes involve complex decisions. This translation can also highlight the observable values quantitative / qualitative data relevant to the simulation objectives, in addition it specifies the kind of behavior to simulate (detailed in section5).

The following metamodel in Figure 4 synthesizes the concepts supporting a CAOM model. An organization is composed of (hierarchical) levels that contain one or multiple groups; in which it contains agents playing roles. An agent plays a single role in one group. An agent is capable of playing the same role in different groups and the same role can be played by several agents. Organization, group, and agents can generate observables (quantitative or qualitative).

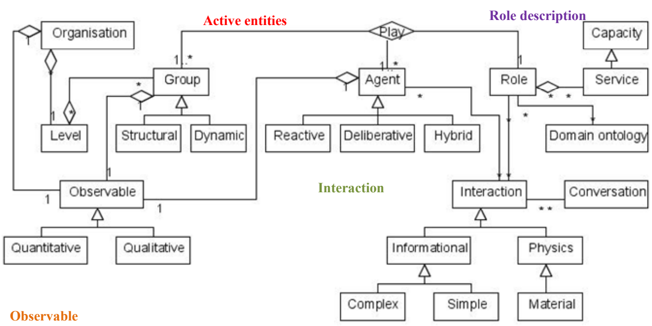

Figure 4: CAOM Metamodel

A role provides services to other roles of the same group, while a service may require capacities (as defined in the domain model). Each role is associated with a (sub) domain ontology. Interactions connecting agents may hold between agent and/or roles. The structural or dynamic nature of groups defined in the CROM model is conserved. Informational and physics interaction subtypes relate to the same distinction (detailed in Table 2).

Type of interaction	Categories of interaction	Graphical notation	Description
Physical	Material	→	Exchange of material between actors is represented by the reactive agents. i.e. material delivery (Stock Truck)
Informational	Simple	⇢	Simple exchange of information to achieve tasks, i.e. allocation of tasks, knowledge sharing.
	Complex	↔	Suppose that agents must coordinate their actions in order to provide all their skills to solve more complex tasks. For example, industrial activities that require a distributed approach, such control systems, design and manufacture of industrial products, distributed control.

Table 2: Interactions type

As a CAOM model is defined by a set of agent, groups and levels (themselves defined by groups and their agents), the structure of the group can be defined as the quadruplet: $G_i=\{A_i,S_i,I_i,T\}$,where A_i is the sets of agents belonging to the group G_i, S_i is the set of the services of the agents, I_i is the set of interactions between the agents, and T denotes the time simulation identifying the time scale {short term, medium term, long term} as defined in the CROM. This model is inspired by AGR [Ferber et al., 2009] and CRIO [Gaud et al., 2008].

The principle translation task is to decide about the role that should be included in the CAOM model. Roles can be combined into one or several agents, according to the kind of behavior which is expected to be studied (simple or "intelligent" machine, workshop global or internal behavior…). Table 3 summarizes different criterion used to decide on the role translation method used for a CAOM model. For example, in the Agent-Actor model "mechanical" roles are described with reactive agents, (implemented in AnyLogic[4]) whereas role with complex behaviors are enacted by deliberative agent (Majorca platform [Labarthe et al., 2007], [Tranvouez et al., 2006]). If a CRIO agent/holon architecture is chosen to support the CAOM model, roles can be played by a hybrid agent.

CROM	CAOM	
	Agent <Type>	Expected behavior (translation criteria)
Roles of the actor	Reactive agent <RA>	If simple behavior is required, a stimuli-response behavior type is sufficient. This can be described later on with a UML state chart diagram. (for example a production machine)
	Deliberative agent <DA>	If decision-making and negotiation is needed then capacities will require a deliberative agent to perceive its environment and other agent behavior. Example: production manager (CROM Example)
	Hybrid agent <HA>	Reactive and deliberative behaviors are required. For example an "intelligent" machine capable of cooperating with other machines when disrupting events while occurring.

Table 3: Actors to agents

The second task of this translation to a CAOM model concerns the transformation of the relationship between the CROM actors. Thus, CROM relationships are transposed in agent world as interactions while keeping their classification (cf. table 3).

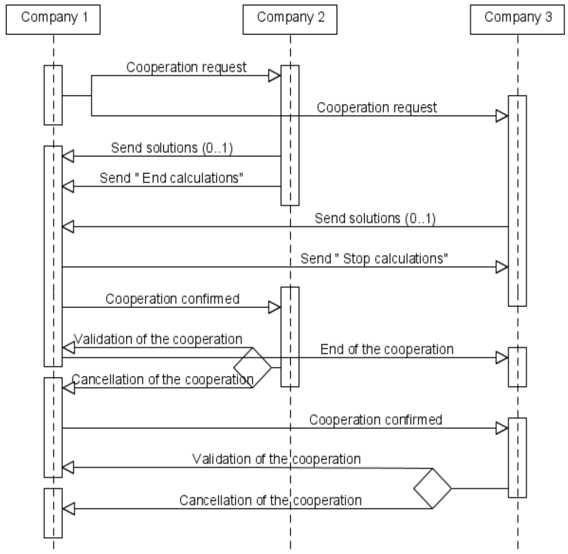

Figure 5: Communication modeling

Figure 5 shows a sequence diagram defined with UML which was used to model the communication between agents.

4.2.2 An Illustrative CAOM model of a SC

The following example illustrates a CAOM model that is used to describe the organizational structure of a given SC. The structure is divided into 3 levels.

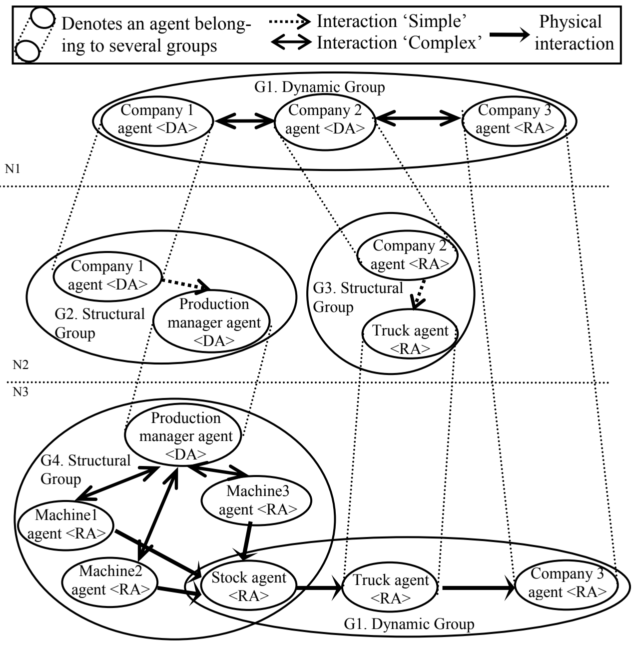

Figure 6: A CAOM of the considered SC

Each level consists of one or more groups (structural or dynamic) of agents. In the first level, *Company 1*, *Company 2* and *Company 3* are three agents connected by an (complex) interaction; each of these companies is represented by a group of agents playing roles at different hierarchical levels. Figure 6, gives a graphical descrip-

[4] www.xjtek.com

tion of a CAOM model with reactive and deliberative agents. In figure 6, the VMI process is described: because of *Company 3* limited behavior, it is represented by a reactive agent; it uses *Company 2* as a truck resource when needed. Physical interaction specifies the flow of information with agent use in order to accomplish their objectives.

5 OPERATIONAL MODELING

This section provides a description on the necessary tasks used to create the *OPerational Agent Model* (OPAM). An approach to transform an agent conceptual model (CAOM) to an operational model is presented. We also describe the general requirement of the simulation architecture, which much support a reactive and a cognitive agent environment as well as their interoperability.

5.1 Operational Agent Model (OPAM)

The operational model provides a solution for implementing the conceptual model (CAOM). This step has led to the development of an operational model agent including the choice of agent architectures.

For the representation of agents and their behavior at the operational level, we propose a modeling approach which allows agent type differentiation. This approach, guided by the nature of the chosen observables, is based on two software "agent" environment: one for reactive agents and another for the cognitive agents.

Agents present in the cognitive environment act independently to achieve their goals. They have an explicit representation of the environment and have reasoning abilities. The cognitive agents can play several roles in the multi-agent system by the implementation of multiple plans. Agents in the reactive environment act in response to environmental stimuli.

5.1.1 Specifying the behavior of agents

The development of the model OPAM consists of a comprehensive description of the behavior of agents and their interactions content. The choice of an agent architecture has been made at the conceptual level to specify how they could ensure roles and services (i.e. plans which allow them to perform this role).

Each conceptual agent is represented by an agent model chosen among two software architectures: reactive and cognitive. This architectural duality leads us to consider two agent platforms. Each platform manages homogeneous agents' type, with specific simulation capacities and constraints.

The design of the agent business model defines the agent models involved in the multi-agent system. This modeling step includes the specification of the agents and relies on formalisms adapted to each agent type in order to represent the agents' behavior. The behavior of reactive agents is specified using the modeling language AUML (Agent Unified Modeling Language) [Odell et al., 2001]. The cognitive behavior of the agents is specified using the RCA formalism (representation of the behavior of agents [Tranvouez and Ferrarini, 2006]).

5.1.2 Specifying the interactions between agents

The interaction between cognitive and reactive agents within the Operational Agent Model is formalized by sending and receiving messages. The refinement of the interactions description involves characterizing the type (message versus signal) and the content of these messages. The interactions between agents can be declined by three possibilities: (i) the interaction between cognitive agents, (ii) the interaction between cognitive agents and reactive agents, defined in terms of accountability relationships, and (iii) the interactions between reactive agents.

The interactions between cognitive agents are defined as messages. The interaction between cognitive agents and reactive agents requires a transformation of the relevant messages (using sequence diagrams defined in UML) between the two chosen environments for the development and implementation of the system. As for the interactions between the reactive agents, it is identified in terms of signals. Table 4 details the different types of interactions.

Type of interaction	Graphical notation	Description
Message	⟷	Cognitive agents communicate by exchanging messages. Each cognitive agent is associated with an instance of the Jess inference engine.
Signal	········►	Reactive agents interact by exchanging signals.

Table 4: Interaction description

5.1.3 An Illustrative OPAM model of a SC

Figure 7 is an example where the structure is divided into two environments, one environment for cognitive agents and the other for reactive agents.

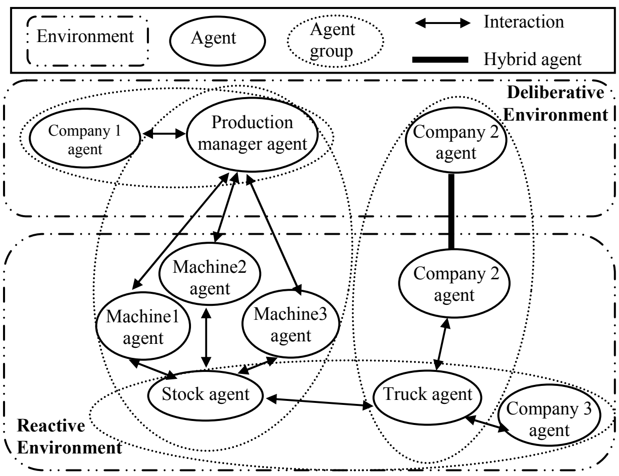

Figure 7: An OPAM model of SCs

For example, agent *Company 1* and the *Production manager* agents are connected by a message interaction type (deliberative environment). Agent *Company 2* is a hybrid agent since he plays two different roles, one cognitive in the cognitive environment and other reactive in the reactive environment.

5.2 OPAM Implementation and Simulation

The implementation and simulation of the OPAM model assumes the existence of a software architecture that supports implementation and distributed simulation of the two kind of agents (cognitive and reactive). In addition, it ensures the integrity of this distributed simulation while providing the desired simulation data (observable).

First we present the distributed simulation based on two different software environments retained for successful implementation of cognitive agents and reactive agents. Then, we describe the general requirements of a simulation infrastructure integrating these two specific environments and supporting their interoperability.

5.2.1 Distributed simulation for cognitive and reactive agents

For the development of cognitive agents, we have chosen Majorca, a platform developed at the LSIS laboratory that provides a development environment for multi-agent systems based on behavioral plans. Majorca is supported on the Jade multi-agents platform and use Jess inference engine to implement complex agent behaviors.

The implementation phase of reactive agents requires the programming of the statechart diagrams. We have selected the Anylogic simulation software, dedicated for the development and the simulation of complex discrete systems, continuous and hybrid models. Anylogic offers an environment of discrete event simulation developed in Java and includes statechart modeling. The choice of Anylogic was further motivated by the possibilities offered in terms of interoperability (Java classes, databases, etc...). It incorporates explicitly the notion of time (event agenda, scheduler…) and has already been used successfully in our former works to connect cognitive and reactive agents [Labarthe et. Al 2007].

The proposed agent-based simulation environment is based primarily on interoperability between the two platforms. It also defines user interfaces allowing users to design models, scenarios, visualize the evolution of the system and its components and finally to exploit simulation results. Compared to the architecture previously proposed in LSIS [Labarthe et al., 2007], coupling between two simulation platforms is not be hard coded ie defined and developed during implementation. The coupling will ensure the interoperability between two different environments in a more generic way, by defining parameterized software bridges with values stemming from the conceptual models.

5.2.2 Simulation architecture for OPAM simulation

Based on the above, we propose a simulation architecture environment consisting of the following elements: (i) Majorca for the implementation of each cognitive agents of the operational model, (ii) Anylogic for the execution of reactive agents, (iii) a Mediator module ensuring the interoperability, and (iv) a database to record the parameters of the backup scenarios and simulation results. The Mediator module provides a set of essential services to interoperability. The most interesting services are: (i) Management of organizational model ie management of the groups and roles dynamics; (ii) Transformation management, which enables agent communication interoperability between different platforms (e.g. message processing vs. signal and signal vs. message); (iii) Time and event management: as different simulation paradigms and groups time horizon are involved, this module ensures the synchronization between different platforms.

This architecture can be considered as an organizational architecture for the implementation and simulation of complex systems. Compared to the previous architecture proposed in LSIS, significant improvements are planned in term of conceptual support (observables and organization) and genericity. Among those, one or several mediators have been defined to ensure the interoperability between two or more simulation platforms, including time management (on which synchronization between different platforms is based) as well as the management of the organizational model.

6 CONCLUSION

This paper presented an organizational oriented methodological framework, which permits modeling and simulation of SC organizational aspects. It allows observables of different level of detail while reproducing the SC behavior according to desired observables. This methodological framework is structured according to two main abstraction levels: a conceptual level and an operational level. The conceptual level, it is composed of a *Conceptual Role Organizational Model* (CROM), which is refined into a *Conceptual Agent Organizational Model* (CAOM). As for the operational level, it is mainly composed of the *OPerational Agent Model* (OPAM). This framework has to allow the study of the impact of a specific SC organizational structure and its related management policies on SC performance. Based on a SC expert modeling of a particular SC, an organization/role oriented (CROM) and an agent-oriented (CAOM) conceptual model helps in designing a simulation model, which will reproduce the SC global and local behavior. These conceptual models are defined independently of particular agent architecture or even on specific software architecture but propose transitional steps to guide their development.

Current work is first looking forward at defining translation rules from CROM to CAOM model taking into account the type and level of details of desired observables while respecting the organization structure and the temporal constraints in which different time horizons produce. Then the translation rules from CAOM to OPAM model defining the type of the agents (cognitive or reactive) are defined.

Future work will propose an open software architecture supporting the OPAM implementation and simulation. This architecture will integrate two different simulation platforms permitting a distributed simulation, based on two specific simulation environments, one for cognitive agents and one for reactive agents, and using a mediator software module and a database to support their interoperability.

REFERENCES

Antonio, L., D'Amours, S., Frayret, J.M., "A methodological framework for the analysis of agent-based supply chain planning simulations", *SpringSim '08: Proceedings of the 2008 spring simulation multiconference,* Society for Computer Simulation International San Diego, CA, USA, 2008.

Bruekner S., Baumgaertel H., Parunak HVD, Vanderbok and Wilke . "Agent Models of Supply Network Dynamics: Analysis, Design and Operation, in The Practice of Supply Chain" *Management: Where Theory and Application converge, Harrison, Lee and Neal Eds., 2003.*

Espinasse, B., Serment, J., Tranvouez, E., "An Agent Integration Infrastructure for the Development of Environmental Decision Support Systems based on Simulation", in: *AIS-CMS International modeling and simulation multi-conference*, Buenos Aires - Argentina. ISBN 978-2-9520712-6-0, 2007.

Ferber, J. & al. Towards an integral approach of organizations in multi-agents systems: the MASQ approach. *Semantics and dynamics of organizational models in Virginia Dignum*, 2009.

Gaud, N., Galland, S., Koukam, A., Towards a Multilevel Simulation Approach based on Holonic Multi-agent. *Published in the 10th International Conference on Computer Modeling and Simulation (EUROSIM/ UKSiM'08)*, pp. 180–185, England. April 1–3, 2008.

Hubner, J.F., Vercouter, L., Boissier, O., Instrumenting Multi-Agent Organizations with Artifacts to Support Reputation Processes, Sixth European Workshop on Multi-Agent Systems, Bath, UK, 18-19 December 2008.

Hubner, J. F., Sichman, J. S. & Boissier, O. Developing organized multi-agent systems using the MOISE. *International Journal of Agent-Oriented Software Engineering*, 1(3/4), 370-395, 2007

Labarthe, O., Espinasse, B. , Ferrarini, A., Montreuil B., Toward a Methodological Framework for Agent-Based Modeling and Simulation of Supply Chains in a Mass Customization Context, in: *Simulation Modeling Practice and Theory International Journal (SIMPAT)*, vol. 15, n° 2, pp. 113-136, February 2007.

Labarthe, O., Montreuil, B., Espinasse, B. , Ferrarini, A., "Modélisation multi-agents pour la simulation de chaînes logistiques de type personnalisation de masse*", MOSIM 04, Nantes, France*, septembre 2004

Monteiro T., Anciaux D., Espinasse B., Ferrarini A., Labarthe O., Roy D. , "Chapter 6. The Interest of Agents for Supply Chain Simulation", *in: Wiley-ISTE (Ed.), ``Simulation for Supply Chain Management'', C. Thierry – A. Thomas – G. Bel*, septembre 2008.

Ounnar, F., Archimède, B., Pujo, P., Charbonnaud, P., HLA Distributed Simulation Approaches for Supply Chain", in: *Hermès Science Europe Ltd (Ed.), ''Simulation for Supply Chain Management''*, Hermès Science Europe Ltd, 2008.

Odell, J., Parunak, H.V.D. and Bauer, B. 'Representing agent interaction protocols in UML', *Proceedings of the First International Workshop on Agent- Oriented Software Engineering, CIANCARINI*, P. and WOOLDRIDGE, M. (Eds), 2001.

Govindu, R., Chinnam, R.B., MASCF: A generic process-centered methodological framework for analysis and design of multi-agent supply chain systems. Pergamon Press, Inc., NY, USA, 2007.

Rao A. S. and M. P. Gorgeff, 1991. Modelling rational agents within BDI-Architecture.in J. Allen & al Ed., *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning.* San Mateo, USA, Morgan Kaufmann, Pub, p. 473-484.

SCC - Supply Chain council, "Supply-Chain Operations Reference-model", 2006.

Shen, W., Hao, Q., Yoon, H. J. and Norrie, D. H. 'Applications of agent-based systems in intelligent manufacturing: An updated review'*, Advanced Engineering Informatics*, vol. 20, pp. 415-431, 2006.

Tranvouez, E., Ferrarini A., Espinasse B., Cooperative Disruption Management In Industrial Systems: A Multiagent Approach, *12th IFAC Symposium on Information Control Problems in Manufacturing - INCOM'2006*, EMSE, Saint Etienne, Mai 2006.

Tranvouez, E., Ferrarini, A., "MultiAgent Modelling of Cooperative Disruption Management in Supply Chains", in: IEEE – *International Conference on Service System and Service Management (ICSSSM'06)*, Troyes, France, October 2006.

Vangheluwe, H., et al. An introduction to multi-paradigm modelling and simulation. School of Computer Science, McGill University, Montréal, Canada, 2002.

Zambonelli, F., Jennings N., Wooldridge, M.. Developing multi-agent systems: the GAIA methodology. ACM Trans. on Software Engineering and Methodology, 12(3), 2003.