PAAN: Partial Agreement Negotiation Network based on Intelligent Agents in Crisis Situation

Ayda Kaddouci, Hayfa Zgaya, Slim Hammadi and Francis Bretaudeau

Abstract— The aim of this paper is to present a multi-agent based approach for negotiation in crisis management. We propose to harness the potential of the multi-agent system (MAS) technology for constructing a framework of cooperation agents that are capable of delivering an optimal solution for crisis.
Supply Chain study is adopted more and more for the companies' competitiveness development. Our industrial partner EADS (European Aeronautic Defence and Space Company) handles a logistic flows demonstrator for crisis management, developed by our research team. A Multi-Agent architecture is planned to design a distributed supply chain. In this paper, we propose to adopt an advanced interaction between the autonomous entities. Therefore, we propose a multi agent based architecture for crisis management supply chain. A new form of negotiation is presented to avoid, in a crisis situation, the stock-out by balancing the resources provisions throughout the system.

Keywords— Multi-Agent System, Distributed Supply Chain, Negotiation, Protocol, Crisis Management.

I. Introduction

The Supply Chain (SC) seems to be a promising answer for the organizations success. A SC represents the links connecting the final customer to the first level supplier, it is thus necessary to develop the SC management by optimizing flows going from the supplier to the customer and vice-versa(e.g. information, physical and goods flows). However, a phenomenon of orders' variability can appear within a SC, expressed by a demand amplification. This phenomenon, recognized by J.W.Forrester [1] and named thereafter by Lee et al. [2,3] the Bullwhip Effect, reduces the SC agility and incurs costs due to higher inventory levels. In our work, we focus on a special kind of SC: a distributed Crisis Management SC (CMSC) characterized by a hierarchical structure and an expected high disturbance impact.

It is never easy to anticipate the evolution of a supply chain. Consequently, to integrate the disturbances as a parameter into the study of the chain limits its vulnerability. In this paper we propose a new form of negotiation to avoid, in a crisis situation, the stock-out by balancing the resources provisions throughout the system. Besides, the proposed model takes into account the disturbance impact to decide on the used protocol. To situate our solution, we start this paper with a state of the art about the negotiation techniques in multi-agent systems. Then we present in paragraph 3 the multi-agent supply chain system. In the 4th paragraph, we address the proposed negotiation form for the provision balancing. Finally, experimentations in paragraph 5 show the contributions of the proposed model.

Manuscript received January 14, 2010.

Ayda Kaddouci, Hayfa Zgaya and Slim Hammadi are with the Laboratoire d'Automatique, Génie Informatique et Signal –LAGIS UMR CNRS, Ecole Centrale de Lille, Villeneuve d'Ascq 59851 France (e-mail: ayda.kaddouci@ec-lille.fr; hayfa.zgaya@ec-lille.fr; slim.hammadi@ec-lille.fr).

Francis Bretaudeau is with the Logistics Department EADS Defense and Security Systems, Elancourt Saint Quentin en Yvelines, France (e-mail: francis.bretaudeau@eads.com).

II. Related Works

In this section, we will briefly describe related works about the traditional supply chain management and agent negotiation.

A. Multi Agent in Supply Chain

Supply chain management represents a competitive advantage that companies try to maintain.

Traditionally, centralized planning systems have been used for production planning in a single company. Offering a complete view of the production activities, they usually use optimization algorithms to find the best production planning solutions. In a distributed context like supply chains, where different partners work together to deliver goods to final customers, planning problems become rapidly too complex to solve centrally.

Supply chain is considered as an integrated approach for the monitoring of the flows going from the supplier to the customer. Indeed, SCM aims to efficiently integrate suppliers, manufacturers, warehouses, and retailers, not merely to ensure that merchandise is produced and distributed in the appropriate quantities, to the right locations, and at the right time, but also to minimize system wide costs while satisfying customer requirements. However, this integration is difficult for two primary reasons:

- Different supply chain facilities may have different, possibly conflicting, objectives.
- Supply chains are dynamic systems that evolve over time.[18]

The integrated SC or supply network, goes beyond the traditional concept of logistics and flow of materials, and emphasizes the need of the interaction between the several actors of the chain during the creation of customer service.[19]

In order to increase the supply chain effectiveness, and so reduce total cost, efficient communication and rigorous negotiation mechanism are required.

Software agent technology provides the environment that facilitates the assimilation of the entire supply chain as a net of independent entities.

The intelligent agents co-operate and compete to reach the desired objectives designed by their owners. The ability of multi agent systems to offer robustness and efficiency, to allow inter-operability and to apprehend the dynamic nature of the supply chain is a major asset for representing the different actors.

To coordinate different supply chain entities and respond efficiently to the customer's needs, negotiation decisions turned out to be very crucial.

In fact, negotiation techniques are used to overcome conflicts and coalitions, and to come to an agreement among agents, instead of persuading them to accept a ready solution.

Negotiation is the most appropriate contribution form to the conflicts resolution because it allows the decentralization of the control by tolerating equitable authorities among agents [6]. There are many classes of negotiations that can be classified in two different ways: according to the agent's self-interest: cooperative or competitive negotiation or according to the adopted form (the protocol).

B. Cooperative and competitive negotiation

There are two main kinds of negotiations: competitive and cooperative negotiations: A negotiation is competitive when concerned agents try to increase their own local utility. However, a negotiation is cooperative when concerned agents try to reach the maximum global utility, which takes into account all their activities. This kind of negotiation can be considered as a distributed research process.

C. Different forms of negotiation

This classification organizes the negotiation according the form that it can take, from the most basic form to the most complex one:

- The "take it or leave it offer": it corresponds to the most basic form of negotiation. An agent makes a proposition in only one loop. This proposition will be accepted or refused without a counter-proposal or renegotiation.

- Voting systems: the purpose is to propose an alternative then to receive the pros and the cons votes for this alternative.

- Auctions: The goal is to sail goods with best prices. The manager begins the negotiation with a possible initial price, and then the proposed offer increases or decreases until one of contractors accepts the current reached price.

- The Contract Net Protocol: corresponds to a higher form of negotiation, proposed by Smith in 1980 [8] for the tasks assignments in a network. Almost all negotiation protocols are based on this form of interaction: when a manager wants to delegate a task, he sends out a call-for-proposal to find the contractors within his network who can help him to make this task. So he collects the different received offers and chooses the most appropriate one to him.

- Multi-attribute negotiation: this form takes into consideration several items on which the agents negotiate.

- Multi-level negotiation : the contract corresponding to this form of negotiation is decomposed into several dependent sub-contracts, which are considered sequentially.

- Combined negotiation: correspond to a set of independent negotiations, which can take several forms. However, the negotiated items are interdependent.

- Argumentation-based negotiation: consists on the modification of negotiating agents' believes in order to have the same intentions with the argumentative agent.

Given the context of competitive multi-agent coordination in a highly dynamic environment, one of the main problems is to solve deadlock situations. In the following sub-section, we present the multi-agent based approach for crisis management.

III. The Multi-Agent Supply Chain System

Disasters, like earthquakes, floods, etc. can cause large number of deaths, injured and homeless. Appropriate responses, in terms of allocating resources, are needed to handle the effects of disasters.

Crisis management relies upon geospatial information to represent the geographical distribution of events, its causes, stricken people and infrastructure, and available resources.[9]

The CMSC holds the same objective as a commercial logistic chain: satisfy the resultant needs of the effects of the crisis. Those specific logistics needs are: help of the victims, reconstruction of minimum infrastructure, providing food, water, medical support, etc. Attaining these goals requires the involvement of different and separate entities. Indeed, the CMSC is composed of several dynamic and geographically distributed areas. The sitting of these areas is a strategic decision of great importance, taken following a crisis, and considering certain conditions of distance, ease of access and political stability to be valid and effective. Each area or zone must cooperate in order to satisfy the needs of the whole system.

Our CMSC is represented by the various links below:

- Warehouses: which manage the stocks of material to insure the routing of the necessary quantities the D-Day,
- Z1: it is the zone where all the products must be gathered together for the routing,
- Z2: it is the base which distributes products towards the various demanding zones,
- Z3: it is the zone which delivers directly products to the stricken zones.

The ability of multi agent systems to offer robustness and efficiency, to allow inter-operability and to apprehend the dynamic nature of the supply chain is an undeniable advantage of being a model that closely represents the CMSC and fits the reality.

In fact, modern intelligent systems contain agents (intelligent computerized assistants) that are capable of acting autonomously, cooperatively, and collaboratively to achieve a collective goal. Intelligent agents can act on behalf of humans and assist them in executing complex tasks. Various researchers [10]-[16]-[17] indicate that multi-agent system seem to be applicable for applications that can be naturally modeled as societies of interacting autonomous entities [9].
Our proposition is to consider each actor of the CMSC as an autonomous agent, able to exchange information with other actors. In our supply chain, actors are many and varied and many models are possible. However, they all involve modeling the different areas of the supply chain through one or more agents. Thus, information exchange and goods' trade between areas are primarily the result of a direct or indirect communication between those agents.
The CMSC is an L-levels SC links; from the provisions warehouse for routing Z1 (exclusive first level) to several disaster zones ZL. All other zones are of level i with 1<i<L.
So for a given zone Zi, a downstream zone is of level i+1: Zi+1 and its upstream zone is of level i-1: Zi-1.
The retro logistic is not allowed within our CMSC, so the matter flow goes from the upstream to the downstream nodes. However, the data flow can take place in the two directions.
The hierarchical feature between the various entities characterizes our multi-zone logistic system. So there is an agent responsible of each zone representing it, we call this agent: a zone-agent. Each zone-agent can communicate only with another zone-agent that is hierarchically higher/lower to him (an upstream/downstream zone-agent) or with another zone-agent from the same hierarchical level.
For example, let's consider a 4-level CMSC:

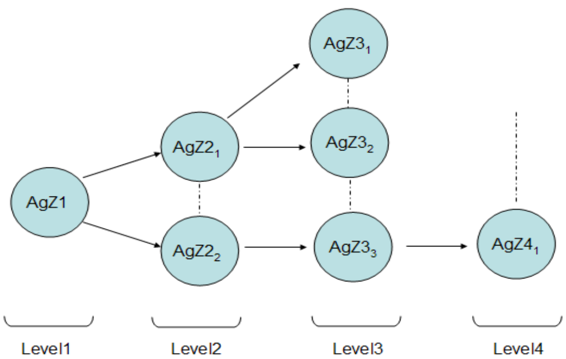

Fig. 1 a 4-level crisis management supply chain

- AgZ1: corresponds to the Z1 zone-agent (level 1) who is the direct AgZ2i (level 2) hierarchical superior,

- $AgZ2_i$: corresponds to the $Z2_i$ zone-agent (1≤i≤N) who can interact with AgZ1 or with any another agent $AgZ2_{i'}$ (1≤i'≤N and i'≠i). This agent is the direct $AgZ3_{i,j}$ (level 3) hierarchical superior,

- $AgZ3_{i,j}$: corresponds to the $Z3_{i,j}$ zone-agent (1≤i≤N and 1≤j≤M) who can interact with AgZ2i or with any another agent AgZ3i,j' (1≤j'≤M and j'≠j). This agent is the direct $AgZ4_{i,j,k}$ (level 4) hierarchical superior,

- $AgZ4_{i,j,k}$: corresponds to the $Z4_{i,j,k}$ zone-agent (1≤i≤N , 1≤j≤M and 1≤k≤P) who can interact with $AgZ3_{i,j}$ or with any another agent $AgZ4_{i,j,k'}$ (1≤k'≤P and k'≠k).

The hierarchical relationship between the different zone-agents of the CMSC is verified thanks to a social capacity function. So the zone-agent, who receives the message, can check if the sender zone-agent is authorized to communicate with him.
Other agents can then intervene, and will help to smooth exchange of information and resources.

IV. Description Of The Different Agents Of Our Architecture

A set of agents was developed for solving coordination problems of different actors in the CMSC. The proposed MAS requires delegation of task to various agents in the system. In the following section, we will describe the organization of different agents of the system.

A. *Zone agent*

Each zone of our CMSC is represented by an agent. It is being used for the following purposes:
• providing names and addresses of superior and subordinate areas,
• providing the current stock levels and a forecast upon 7 days,
• providing the future needs of the area, and its subordinates,
• providing the level of safety stock
• providing the number of population on the zone.

B. *GUI agent*

The GUI agent has been designed for providing an interface. It is being used for the following purposes:
• Dialoguing with others agents on the platform and passing them information,
• Displaying the information related to each area or zone,
• Feeding the system by data: stock levels, prevision and delivery periods, resources…
• Displaying the feedbacks about the supply chain.

C. *Weather agent*

This agent gives information related to the environment in a specific area; three types of data were identified: the temperature, the humidity and the number of civilian to be provided with supplies in the concerned area.
The Weather agent provides an estimate of the data for the actual day of simulation, and the next 6 days.

D. *Need Estimating agent*

The Need Estimating Agent (NEA) is a tool for decision support that is supposed to indicate to a zone-agent what it will need; using all the data that the zone-agent can provide.
A zone-agent is a supplier for their sub agents, but it is also a consumer for its own needs because there are people to feed, engines to fill, etc. The NEA provides an estimation of these needs, based on data collected from the different agents (Weather agent, Zone agent...). It mainly works using fuzzy logic calculation. [5]

E. Posts Coordinator agent and Consumption agent

These agents role is to ensure the smooth functioning of the supply chain.

The Posts Coordinator agent handles haulage and informs areas they just received a package. When a zone decides to send goods to another zone, the Posts Coordinator agent is informed. Then he waits a certain number of days, which correspond to delivery period, then notifies the receiving area that a package has arrived. Finally, this agent can be assimilated to logistics teams that manage trucks, boats and other vehicles transporting resources

The Consumption agent provides daily each zone with the quantity of goods consumed by the local population indicated in this area.

V. The Proposed Negotiation Model

When there is a pressing need of provisions and the CMSC is affected by some perturbations, we propose to start a cooperative negotiation process, which propagates throughout the hierarchical structure: at most from the first node until the last one. The goal is to balance resources in the best way, along all the nodes within the different levels. In this context, we propose to adopt an advanced form of interaction described and argued in what follow.

According to Jennings et al. [12], a negotiation process design requires the elaboration of three main features: the negotiation protocol, which corresponds to the set of rules controlling the interaction, the items on which the agents negotiate and finally the decisional model which correspond to the strategic layer of the negotiation. These three features depend on the negotiation context and have to be considered within the negotiating agent architecture.

A. Comparison and argumentation

From the classification that we presented in the first section, we propose to situate our problem by comparing the different existent forms of negotiation according to the fundamental interaction framework of the CMSC.

Table. I comparison with existent forms of negotiation

Negotiation form	Similarities
Voting systems	No
Auctions	No
The Contract Net Protocol	Basic steps
Multi-attribute negotiation	Several items
Multi-level negotiation	Sub-contracts but simultaneously
Combined negotiation	No
Argumentation-based negotiation	Argumentation: the global CMSC survival

- Voting systems: This form of negotiation needs only one intervention from participants to conclude the vote procedure. So it cannot be useful to our CMSC interaction features,

- Auctions: the negotiation result is satisfactory only for the agents who draw up the contract. All the other participants are kept in the background. In addition, only one item is negotiated. This is not a suitable form because in our CMSC we have to negotiate several vital resources and we aim to satisfy all distributed nodes,

-The Contract Net Protocol (CNP): this form facilitates the distributed control of the cooperative tasks execution. Besides, at any time, any agent can be an Initiator (manager), a Participant (contractor) or both. The interaction framework of our CMSC basically uses the 4 steps-CNP as follows: 1-The initiator sends the resources needs; 2-Each participant reviews the needs and bids on feasible ones; 3-Initiator chooses interesting bids and awards a contract to each concerned Participants; 4-Initiator rejects other bids.

- Multi-attribute negotiation: in the CMSC, we have to dispatch several kinds of resources, which have to be negotiated simultaneously,

- Multi-level negotiation: the contract is decomposed into dependent sub-contracts, which are considered sequentially. In our CMSC, an Initiator need to balance resources among the different Participants, so the global contract is shared into sub contract according to each pair of Initiator/Participant. These sub-contracts are dependent because the initiator has to find an efficient balancing among all agents. However, to reach an efficient balancing, we have to manage simultaneously all current sub-contracts,

- Combined negotiation: there is no similarity because for this form of negotiation, contracts are independent,

- Argumentation-based negotiation: The Initiator can make a counter-proposal trying to convince Participants to change their previous offers. So knowing that we adopt a cooperative negotiation in our CMSC interaction framework, Initiator agents use the global SC survival argumentation.

So, in our CMSC interaction framework we adopt among four existent negotiation forms, the “simultaneous” multi-level negotiation. To manage this simultaneity, we propose to add the “partial” aspect. In other word, we propose that an Initiator agent, who manages simultaneously several sub-contracts, can accept each sub contract totally or partially within each negotiation loop. So we call the new form of negotiation: the Partial Agreement Negotiation Network (PANN).

B. The Negotiating Agent Protocol

The adopted negotiating agent architecture is composed of three layers; inspired from [4]:

1. Communication Layer: corresponds to the interaction layer of the architecture, it is responsible for receiving and sending messages between agents;
2. Control Layer: corresponds to the negotiating agent behaviors specified by an activity UML sate chart;
3. Reasoning Layer: corresponds to the decision-making part of the negotiating agent and interacts with his Knowledge Base module.

A negotiation process, Identified by NegId, is composed of:

- Initiators of the negotiation who start the process. An Initiator is noted by $Init_i$ ($1 \leq i \leq I$),
- Participants who contribute to this negotiation. A Participant is noted by $Part_j$ ($1 \leq j \leq P$),

Objects of the negotiation: limited resources on which the negotiation members (Initiators and Participants) negotiate. A resource is noted by r_i ($1 \leq i \leq R$).

According to the CMSC perturbation impact, we distinguish two kinds of PANN[1] protocols, corresponding to the Communication Layer: the Help-One-To-Many (HOTM) and the Help-Many-To-Many (HMTM) protocols. The decision on which protocol to be used depends on the agent-zone Reasoning Layer.

C. The Communication Layer

∗ *Help One-To-Many Protocol (HOTM)*

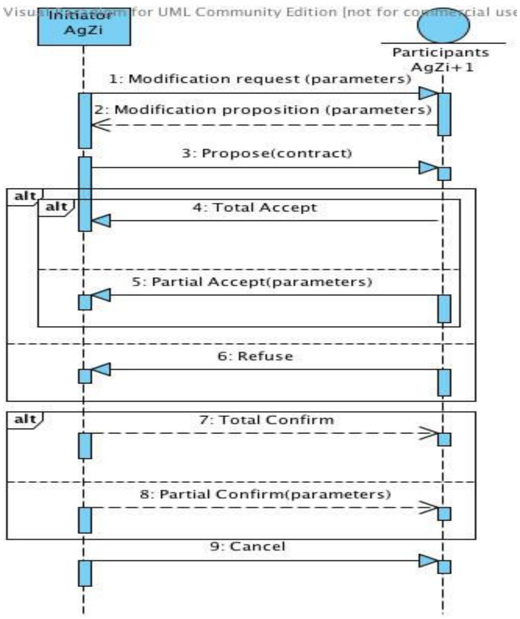

Fig. 2 the HOTM protocol

We focus here on the case of a single Initiator for hierarchical reasons. The HOTM Initiator is a single agent-zone AgZi, who realizes that he cannot satisfy all the demands including his own needs. So he starts the negotiation waiting for some desistance from his correspondent downstream agent-zones, the Participants (agents-zones AgZi+1). So the first agent-zone can be an Initiator, the last agent-zone can be a Participant and all the other agent-zones can be, even simultaneously, both; at the same time an Initiator for downstream agent-zones and a Participant for his single upstream agent-zone.

This protocol is described as follows:

- Modification Request: If the Initiator realizes that he cannot satisfy all his subordinate zones demands before some period of time Δt (new supply delay), he informs all the subordinate agent-zones about the situation proposing them to renounce to their demands if they can wait for an additional period of time. In other words, as soon as an upstream agent-zone is not able to response to some resources demands, the control Layer is activated by a modification demand and an "output event" starts the HOTM protocol,

- Modification Proposition: each agent-zone Participant $Part_j$ ($1 \leq j \leq P$) sends his Emergency degree for each resource r_i to the Initiator: $E_d(r_i, Part_j)$ $\forall i$ ($1 \leq i \leq R$). This corresponds to an "input event" by the Communication Layer within the Initiator negotiating Agent Architecture.

- Propose (contract): The Initiator sends a new contract expressing the latest provisions quantities balancing, evaluated within the Reasoning layer,

- Accept/Refuse: After estimation of remaining inventories of all the provisions (water, medicines, clothes, etc.), a Participant agent-zone AgZi+1 realizes that he can accept:
  - All the Initiator propositions (Total Accept),
  - A sub-set of the Initiator propositions (Partial Accept). For example, he can accept the given Initiator proposition for clothes but not for water and medicines
  - None proposition (Refuse).

- Confirm: the Participant confirms totally if he still agrees with the accepted proposition and partially if he changed opinion concerning what he already proposed. This change of opinion can be the result of an urgent need occurred locally within the Initiator agent-zone. We notice here that further to a confirmation, the Initiator sends the accepted amounts of provisions. In this case, if there are still some downstream agent-zones (AgZi+1) who still, in real time, need some provisions and the correspondent upstream agent-zone (AgZi) can't satisfy all the demands, the negotiation process loops (go to Modification Request demand). Otherwise, the protocol ends.

- Cancel: the negotiation process can be cancelled (e.g. at the end of authorized negotiating time.

∗ *Help Many-To-Many Protocol (HMTM)*

When the CMSC disturbance impact is very important, the hierarchical aspect of the CMSC is not really respected, so "break the policy" is exceptionally authorized. The idea is to allow an agent-zone AgZi to look for help from any colleague (i.e. any agent-zone from the same level i) or from any upstream agent-zone AgZi-1, who is not necessarily his direct hierarchical superior. So, a HMTM Initiator is an agent-zone AgZi who informs that he needs provisions and waits propositions from any agent- zone AgZi or AgZi-1. The HMTM protocol is described as follows:

- Modification Request: This is an alert message from any agent-zone AgZi (from i^{st} level) to all the other agent-zones AgZi (from the same level i) and to all the agent-zones AgZi-1 (from i-1^{st} level).

- Modification Proposition: any agent zone AgZi/AgZi-1 may propose some resources distribution to the current claimant agent-zone according the posted needs,

[1] Partial Agreement Negotiation Network

- Accept/Refuse: An Initiator can receive many propositions for the same needed resource, so he should make a choice according to some features (e.g. resource routing time). The choice is made within the reasoning negotiating agent architecture layer. After the last received proposition study, an Initiator agent-zone AgZi realizes that he can accept:
  - All the propositions of a Participant for all the resources (Total Accept);
  - A sub-set of a Participant propositions (Partial Accept). For example, he accepts only medicine propositions from level_i and foods and water propositions from level_i-1.
  - None proposition (Refuse).

We notice that a refusal resource proposition can rise from, for example, a considerable latency or an expensive routing. Anyway, this decision-making is done through the Reasoning Layer level.

- Confirm: similar by analogy to the HOTM protocol.

- Cancel: similar to the HOTM protocol.

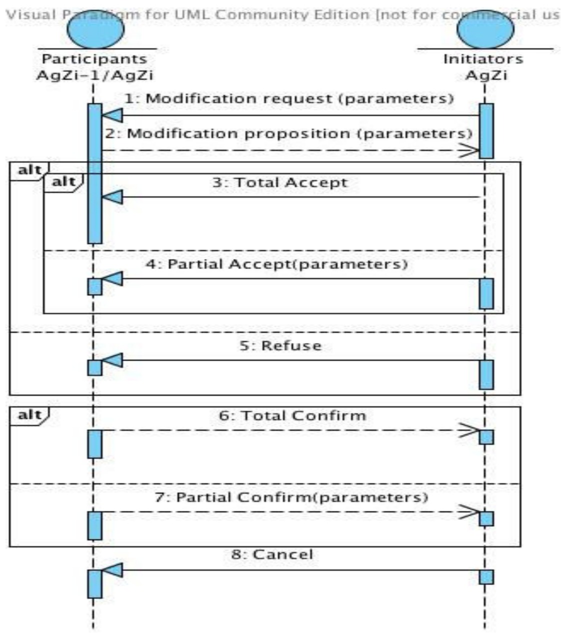

Fig. 3 the HMTM protocol

D. Control Layer

The negotiating agent behavior is specified by an UML statechart. If an agent (identified by *AgId*) dispatches a Call For Proposal (a Modification request) then he corresponds to an Initiator. Thus, he resets a timer and waits for participants' responses (*Modification Propositions*). If the used time (*time_responses*) for the current negotiation process (identified by *NegId*) reaches the waiting delay Δ_{tw} for the responses reception, or if the current Participant number for the negotiation (*number_Part*) realized by the agent-zone equals the received responses number (*number_responses*), the agent *AgId* exploits his reasoning layer to make a decision. If there is a total satisfaction, the negotiation process ends. Otherwise, if the remaining time to the current negotiation is over (*timeNegotiation=0*), the initiator makes a weighted decision according to priorities. If not, the negotiation loops.

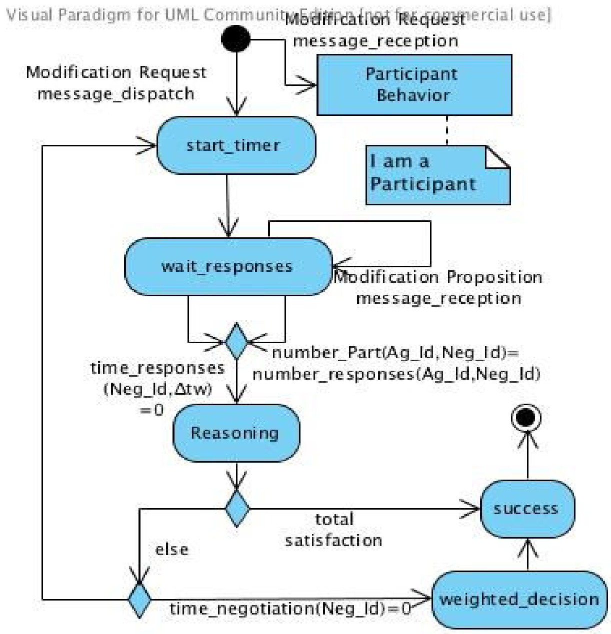

Fig. 4 the negotiating agent behavior

E. Reasoning Layer

As it was previously mentioned, this layer corresponds to the agents reasoning module that generates decisions. In this paper, the proposed reasoning module is based on some decision rules based on estimation of the future needs of resources. In future work, we aim to design this reasoning module according to the Believe-Desire-Intention (BDI) architecture to reach rational interactive MAS.
The reasoning layer is actually made of two sub-layers: an anticipatory layer and an effective layer.

In our work, the goal is to estimate the future resources needs (e.g. water, food, medicine) and so to reduce stock out risks in a disturbed mode, and optimize the provisions distribution throughout the CMSC. Then we adopt a cooperative negotiation model between the concerned agent-zones of the system, in order to balance resources provisions along the CMSC.

∗ *The anticipatory level*

This step concerns the identification of the agents involved in the process of estimation and the relations that link them to each other. For the CMSC, having an accurate prevision of the resources needs in the future is very relevant. The Need Estimating Agent, based on the knowledge about the perceived entities, their goals and coalition memberships and their hierarchical level, will estimate the need in a given resource at a point in the time. The NEA will simulate a model M of the system and so predict the future state of stocks of the CMSC.

The NEA mainly works using fuzzy logic calculation. The zone-agent provides the NEA with the needed data in order to complete the calculation. A human expert is in charge to estimate those data. He has to provide the real information acquired on the field (how many persons to feed, etc). The NEA collects other data such as the average consumption of a given resource (Consumption Agent), the local humidity and temperature on the zone (Weather Agent) and the present population (Zone-Agent).

The fuzzy logic offers a behaviour that is as close as possible to the human reflexes. This is interesting as far as the human factor remains an important parameter to consider in the military hierarchy.

The anticipatory level provides information about the stock levels of each resource and the future estimated consumptions. These information will support the decision making process in the effective level.

∗ *The effective level*

Two decisions are particularly important to the CMSC efficiency: the HOTM and the HMTM Initiators decisions. The HMTM Initiator evaluation depends on some vital features, which decides on the provisions reception quality like the resource routing time. In this paper, we focus on the HOTM Initiator decision which is illustrated as follows:
When the HOTM-Initiator receives the propositions (priorities corresponding to all required resources) of all the Participants or if the waiting delay Δ_{tw} for the responses reception is expired, he has to decide how to balance resources. For that, the HOTM Initiator evaluates an emergency index $E_{index}(r_i,Part_j)$ of each resource r_i ($1 \leq i \leq R$) for a Participant $Part_j$ ($1 \leq j \leq P$):

$$E_{index}(r_i,Part_j) = E_d(r_i,Part_j) \times E_d(Part_j,Init) \quad (1)$$

with:

- $E_d(r_i,Part_j)$: the priority of the resource r_i to the Participant $Part_j$, corresponding to the emergency degree,

- $E_d(Part_j,Init)$: the priority of the Participant $Part_j$ to the Initiator of the negotiation.

The decision made by the Initiator agent to share the local remaining stock depends on the evaluated emergency indexes (weighted amounts), knowing that the remaining stock for a resource r_i for an Initiator negotiating agent is evaluated as follows:

$$reStock(r_i,Init)=stock(r_i,Init)-demand(r_i,Init)\times GE_{index}(r_i,\alpha,Init) \quad (2)$$

with:

- stock(r_i,AgId) corresponds to the local inventory level for the resource r_i related to the agent-zone AgId,

- demand(r_i,AgId): corresponds to the local demand consumption for the resource r_i related to the agent-zone AgId,

- Knowing that ***Ne*** corresponds to the total number of negotiating agents including initiators and participants and $AgId_{k'}$ corresponds to the Initiator identifier so $GE_{index}(r_i,x,AgId_k)$ is the local weighting consumption for the agent-zone identified by $AgId_k$ ($1 \leq k,k' \leq \boldsymbol{Ne}$) and according to the resource $\boldsymbol{r_i}$. In other words, this expression evaluates the importance degree of the $AgId_k$ agent-zone local consumption, according to the total consumption of all the negotiating agents (if $x=\alpha$ then $a_u=1$) or according to the participants agents only (if $x=\beta$ then $k\neq k'$ and $a_u=0$ if $u=k'$):

$$GE_{index}(r_i,x,AgId_k) = \frac{\mathbf{E_{index}}(\mathbf{r_i}, AgId_k)_{1\leq k\leq Ne}}{\sum_{u=1}^{Ne}(a_u \times \mathbf{E_{index}}(\mathbf{r_i}, AgId_u))} \quad (3)$$

The decision made by the Initiator agent to share the local remaining stock is evaluated by:

$$amount(r_i,Init,Part_j)=reStock(r_i,Init)\times GE_{index}(r_i,\beta, Part_j) \quad (4)$$

In future works, we will use Defeasible Logic [7] to model the adopted negotiation strategies. This approach offers more effective strategy specifications for Negotiating Agents development [11].

VI. Implementation

We are developing our system, with the JADE platform (Java Agent DEveloppement Framework) [13]. JADE is a middleware that allows the execution of a flexible multi-agent systems, it offers an effective transfer of messages between agents to ensure communication between them through the FIPA-ACL language (Agent Communication Language), which meets the FIPA specifications [14]. JADE is written in Java, supports mobility, and it is one of the few existing multi-agent platform which tolerates the integration of Web services. JADE has several interesting features that at least make the process of implementation easier. One of these features is the agent Sniffer that enables user to observe message flow among agents. [15]

VII. Simulation Results

A. Case Study

We simulate a cargo loss within the 2nd day representing a high perturbation impact and we consider a 3-levels CMSC scale model composed of the following hierarchical zones: 1 Z_1 (level 1); 1 Z_2: $Z2_{1,1}$ (level 2) and 3 Z_3: $Z3_{1,1,1}$; $Z3_{1,1,2}$; $Z3_{1,1,3}$ (level 3). Each zone corresponds to a crisis management logistic base represented by an agent-zone. We remind that the goal is for each agent-zone to satisfy not only his resources consumptions but also those of all his subordinates.

∗ *Resources*

Each agent-zone has to manage R=4 resources: r_1: clothes; r_2: water; r_3: medicine and r_4: food. But we show here simulation results corresponding only to the resource r_4 (random choice).

∗ *Prevision and Delivery Periods*

Previsions period: they are done upon seven days: the current day and the six next ones. In other words, each agent-zone has to predict his inventories evolutions upon seven days to identify the needs and thus to plan new resources commands. For the delivery periods, we consider 2 days between 2 zones from the same level. Otherwise, we distinguish 2 kinds of delivery periods:

- The regular mode: evaluated with 5 days for a zone Z3 and with 3 days for a zone Z2. We adopt by default this mode;

- The urgent mode: evaluated with 3 days for a zone Z3 and with 2 days for a zone Z2. We move to this mode when it is really necessary because it is more expensive.

When an agent-zone realizes (prevision) that he will start the security inventory for a given resource at the end of the delivery regular/urgent period, he places an immediate order to command some quantity of that resource. Consequently, just before this agent-zone receives the ordered resource, the correspondent inventory level should be in the environs of the security inventory level. Hence, we notice that an agent-zone does not wait until he starts his security inventory level to place an immediate order.

∗ *Graph Presentation*

For the results elaboration, we illustrate in this paper graphs, representing the inventories evolutions (the y-axis) according to the time, expressed by the number of days (the x-axis). In each graph, the parallel line to the x-axis represents the security inventory level for the given resource. Furthermore, we distinguish 3 different phases corresponding to the graph evolution:

- Phase 1: represents the past inventory evolution until today: day n°11;

- Phase 2: represents the future previsions taking into account the cargos sent by the providers; between day n°11 and day n°14 for the agent-zone AgZ2;

-Phase 3: represents the future previsions of the inventory evolution, according to the future demands not yet sent by the providers; from day n°14 for the agent-zone AgZ2.

B. Case 1: Without Negotiation

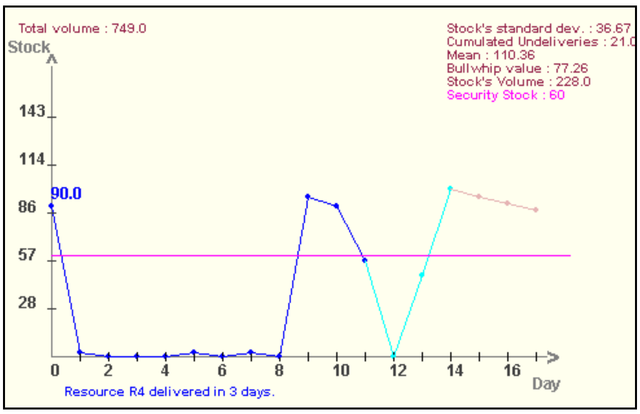

Fig. 5 the AgZ2 inventory evolution without negotiation

When there is no negotiation between agents, we observe that the agent-zone AgZ2 reaches the stock-out condition on several occasions (Fig.5); the cumulated undelivered quantity for the resource r_4 is estimated to 21 units and the Bullwhip Effect value is estimated to 77,26.

C. Case 2: With HOTM Negotiation

∗ *Regular mode*

With the HOTM protocol, the agent-zone AgZ2 avoids just in time the stock-out condition. This comes from a better provisions balancing with the downstream agent-zones. So thanks to this protocol, the situation improves and in addition, the Bullwhip Effect value decreases from 77,26 to 72,16 still thanks to a better provisions balancing.

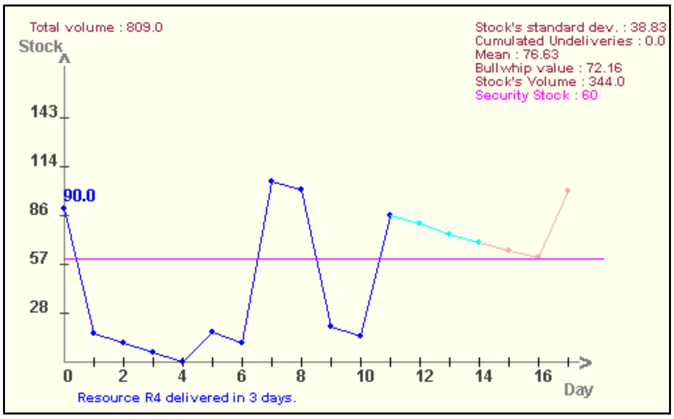

Fig. 6 the AgZ2 inventory evolution with regular mode

∗ *Urgent mode*

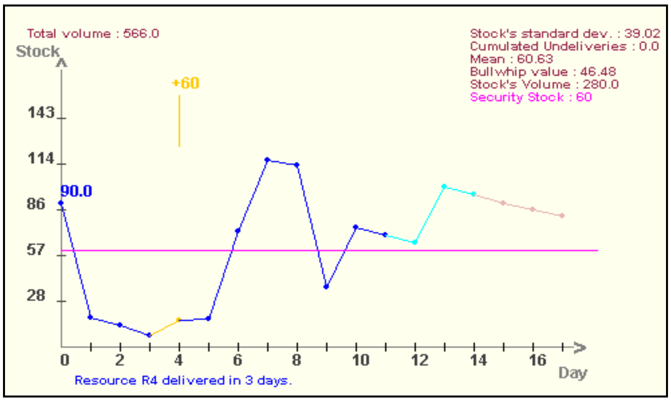

Fig. 7 the AgZ2 inventory evolution with urgent mode

We adopt here the urgent mode: the agent-zone AgZ2 receives 60 units of resource r4 in day 4 from his provider (AgZ1). We observe that the urgent mode improves the stockout risk for the agent-zones AgZ2.

VIII. Conclusion

We are working on a special kind of distributed SC where the different interactive entities are hierarchically related. We proposed for this SC, a multi-agent architecture characterized by independent agent-zones sharing information. In this paper, we focus on the provision balancing in order to avoid the stock-out throughout the CMSC. For that, we propose interaction protocols for the different CMSC agent-zones: the HOTM and the HMTM negotiation protocols.

The experimentation results showed that when we don't consider the negotiation, the stock-out is inevitable. As a future work, we are now testing our negotiation model on a broader version of CMSC. Moreover, we intend to advance the proposed negotiation protocols by modeling the adopted strategies using the Defeasible Logic [7].

References

[1] Forrester, J.W., "Industrial Dynamics," *the M.I.T press-Massachusetts*, Institute of Technology- Cambridge, Massachussetts, 1961.

[2] Hau L. Lee, V. Padmanabhan, Seungjin Whang, "Information Distortion in a supply chain: The Bullwhip effect", *Management Science*, 43(4), 1997, pp. 546-558.

[3] Hau L. Lee, V. Padmanabhan, Seungjin Whang, "The Bullwhip Effect in supply Chains", *Sloan Management Review*, 38(3), Spring 1997, pp. 93-102.

[4] M.J. Wooldridge and N.R. Jennings, "Intelligent agents: Theory and practice", Knowledge Engineering Review 10(2), 1995.

[5] N. Zoghlami and S. Hammadi, "Estimator Agent approach for distributed logistic chain optimization", *In Proceedings of the 50th IEEE International Conference ANIPLA'2006*, Rome (Italy), 13-15 November 2006.

[6] M.H. Verrons, "Genca: un modèle général de négociation de contrats entre agents," PhD theses, University of Sciences and technologies of Lille, LIFL CNRS UMR 8022, France, November 2004

[7] D. Billington, "Defeasible logic is stable", *Journal of Logic and Computation*, 3:pp.370-400,1993.

[8] Reid G. Smith, "The Contract Net Protocol: highlevel communication and control in a distributed problem solver", *IEEE Transactions on computers*, C-29 (12), December 1980, pp. 1104–1113.

[9] A. El-Korany and K. El-Bahnasy, "A multi-agent Cooperative model for crisis management system," *In Proceedings of the 10th WSEAS international conference on Artificial Intelligence, Knowledge Engineering and DataBases AIKED'08,* University of Cambridge, UK, 2008.

[10] H.V.D Parunak, "Agents in overalls: Experiences and issues in the development and deployment of industrial agent-based systems," I*nternational Journal of Cooperative Information Systems,* 9 (3), 209-227, 2000.

[11] M. Dumas, G. Governatori, A.H.M.ter Hofstede and P. Oaks, "A Formal Approach to Negotiating Agents Development", *Electronic Commerce Research and Applications*, Elsevier Science B.V., 2002, 1(2):pp.193-207.

[12] N. R Jennings, P. Faratin, , A. R Lomuscio, S. Parsons, C. Sierra and M. Wooldridge, "Automated Negotiation : Prospects, Methods and Challenges." *Int. Journal of Group Decision and Negotiation*, 10(2) :199–215, 2000.

[13] A. Daknou, H. Zgaya and S. Hammadi, "Towards a tool for scheduling tasks to multiple skills in emergency services," *International Francophone Conference of Automatique CIFA 2008*, Bucharest, Romania.

[14] JADE, Java Agent Development Framework -

[15] A. Daknou, H. Zgaya and S.Hammadi, "Toward a multi-agent model for the care of patients at the emergency department," *In Proceedings of the 10th WSEAS international conference on Mathematical Methods, Computational Techniques and Intelligent Systems MAMECTIS'2008,* Corfu (Greece) 2008.

[16] Wooldridge, M. "An Introduction to Multiagent Systems," *John Wiley and Sons Ltd*, February 2002

[17] Wooldridge, M. and N. R. Jennings. Software Engineering with Agents: Pitfalls and Pratfalls. In IEEE Internet Computing, May/June 1999

[18] C.S. Hsieh and J.W. Hung, "Integration Agent Negotiation and data global consistency forms automatic and none bullwhip effect supply chain," *WSEAS Transactions on Information Science and Applications,* Volume 6, Issue 6, June 2009, pp.1037-1050.

[19] G. Caputo, M. Gallo and G. Guizzi, "Optimization of production plan through simulation techniques", *WSEAS Transactions on Information Science and Applications,* Volume 6, Issue 3, March 2009, pp. 352-362